CHAPTER

2 Fundamentals of CMOS design

Xinghao Chen
CTC Technologies, Endwell, New York

Nur A. Touba
University of Texas, Austin, Texas

ABOUT THIS CHAPTER

The first ***integrated circuit*** (IC), called a ***phase shift oscillator*** composed of one transistor, one capacitor, and three resistors, was created by Jack Kilby of Texas Instruments on September 12, 1958. Today, a typical IC chip can easily contain several hundred millions of transistors and miles of interconnect wires. This ***very large-scale integration*** (VLSI) ability has been enabled by the modern use of the many ***electronic design automation*** (EDA) technologies and applications discussed in this book.

In this chapter, we discuss a few basic and very important concepts of ***complementary metal oxide semiconductor*** (CMOS) technology to aid in the learning process and facilitate greater understanding of the EDA subjects in the subsequent chapters. We first start with an overview of the fundamental integrated-circuit technology and CMOS logic design. Then, we discuss a few more advanced CMOS technologies that can be used to reduce transistor count, increase circuit speed, or reduce power consumption for modern VLSI designs. The physical design aspects, how to translate a CMOS logic design to a CMOS physical design for fabrication, is reviewed and included for completeness. For more in-depth study of specific CMOS technology areas, readers are referred to the various interesting topics thoroughly discussed in the references listed at the end of this chapter.

2.1 INTRODUCTION

The first ***integrated circuit*** (IC) was created by Jack Kilby of Texas Instruments on September 12, 1958. Called a **phase shift oscillator**, the integrated circuit consisted of only one transistor, one capacitor, and three resistors, as shown in Figure 2.1. Since then, IC technology has evolved from TTL (***transistor-transistor logic***) and nMOS to CMOS. Although CMOS was first introduced as an alternative to

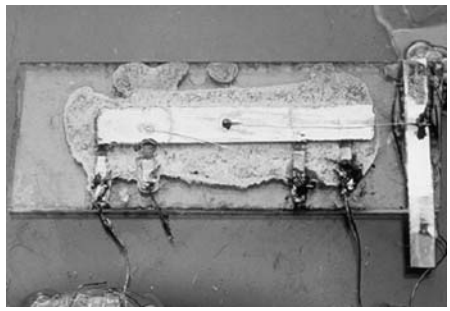


FIGURE 2.1

The first integrated circuit invented by Jack Kilby in 1950 (http://www.ti.com/corp/docs/company/history/timeline/semicon/1950/docs/58ic_kilby.htm, February 8, 2008. Courtesy of Texas Instruments.).

bipolar technologies (such as TTL and ECL), it soon overtook and became the dominant circuit implementation technology. This is because CMOS consumes much less power than TTL and nMOS, as well as the ***very large-scale integration*** (VLSI) capability it provides.

Now, with advanced CMOS process technologies, a chip can contain as many as 2 billion transistors (such as the Intel Quad-Core Itanium Processor, February 5, 2008). CMOS integrated circuits have been the primary digital system implementation technology for consumer electronics, personal, commercial, and enterprise computing systems, as well as electronic systems for scientific exploration.

However, the very large-scale integration ability of CMOS has also created problems that did not seem to be significant in the early days of CMOS technologies. We have seen more and more issues, such as power consumption, thermal effects, small delay defects, cost of test, and validation, dominating the agenda and schedule of a chip design project. Oftentimes, engineers have to make difficult tradeoffs to balance competing design parameters. Aside from providing the reader with fundamental CMOS design and layout principles, this chapter covers some advanced CMOS circuit technologies to assist the reader comprehend the learning process in designing modern VLSI circuits.

2.2 INTEGRATED CIRCUIT TECHNOLOGY

In this section, we first discuss the basic constructs and characteristics of a ***metal oxide semiconductor*** (MOS) transistor (*a.k.a.*, MOS device). Most transistors in digital circuits are switching devices that operate to perform desired Boolean functions. MOS transistors can also be configured as load devices that are used for circuit performance enhancements. Next, transistor equivalency is described, which is a widely used technique for analyzing large and complex circuits. We then discuss

the wire and interconnects that connect the many transistors to form circuits and systems, followed by a discussion of the basic concepts related to noise margin, which is becoming ever more important in low-power applications.

2.2.1 **MOS transistor**

A MOS transistor is a 4-terminal device on a silicon substrate [Martin 2000]. Circuit schematic diagrams often show transistors in 3-terminal symbols, with the assumption that the fourth terminal (known as the substrate terminal) is either grounded or connected to power supply on the basis of the device type. Figure 2.2a shows the dimensions of a MOS transistor, where L is the n-channel length, W is the n-channel width, and t_{OX} is the thickness of the thin oxide layer under the gate. Figure 2.2b shows a cross-section view of a typical **n-channel transistor**. The three terminals of the devices are **Gate**, **Source**, and **Drain**. A fourth terminal connecting the **Substrate** is sometimes provided with devices as well. Common symbols used for n-channel and p-channel transistors are shown in Figure 2.3.

The switching characteristic of a MOS device is determined by its **threshold voltage**, denoted as V_{tn} for an n-channel transistor and V_{tp} for a p-channel transistor. When the effective gate-to-source voltage (V_{GS}) is greater than V_{tn}, a channel will form in a MOS transistor. For an n-channel device, this means $V_{eff} = V_{GS} - V_{tn} > 0$ and $V_{eff} = V_{SG} + V_{tp} > 0$ for a p-channel device, where

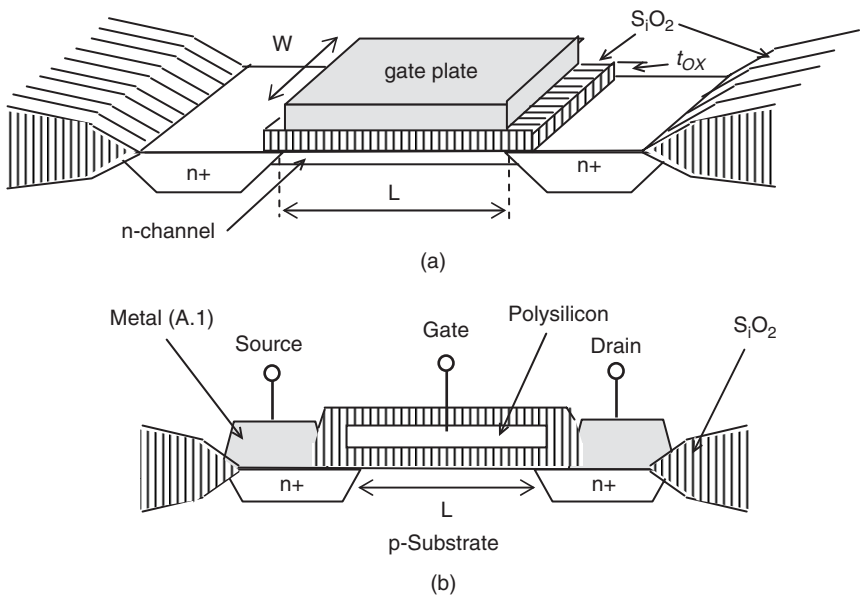


FIGURE 2.2

Illustrations of an n-channel transistor [Martin 2000]: (a) The dimensions of a MOS transistor. (b) A cross-section view of a MOS transistor.

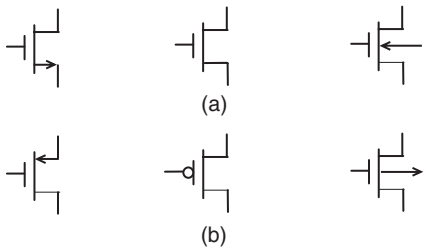


FIGURE 2.3

MOS transistor symbols: (a) For n-channel transistors. (b) For p-channel transistors.

typically $V_{tn} \approx 0.7V$ and $V_{tp} \approx -0.7V$. When the drain-to-source voltage (V_{DS}) is large, the channel current of an n-channel transistor is approximately

$$I_D = \mu_n \cdot C_{OX} \cdot \frac{W_n}{L_n} \cdot \left[(V_{GS} - V_{tn}) \cdot V_{DS} - \frac{V_{DS}^2}{2} \right] \tag{2.1}$$

(where $C_{OX} = \dfrac{\varepsilon_{OX}}{t_{OX}}$ is the gate-oxide capacitance) for $V_{DS} < V_{eff}$ and

$$I_D = \frac{\mu_n \cdot C_{OX}}{2} \cdot \frac{W_n}{L_n} \cdot (V_{GS} - V_{tn})^2 \tag{2.2}$$

for $V_{DS} > V_{eff}$. When V_{DS} is very small, the channel current is approximately

$$I_D = \mu_n \cdot C_{OX} \cdot \frac{W_n}{L_n} \cdot (V_{GS} - V_{tn}) \cdot V_{DS} \tag{2.3}$$

and the channel resistance is approximately

$$r_{ds} = \frac{V_{DS}}{I_D} \approx \frac{L_n}{\mu_n \cdot C_{OX} \cdot W_n \cdot (V_{GS} - V_{tn})} \tag{2.4}$$

Equations 2.1 and 2.2 are known as **large-signal equations**, whereas Equations 2.3 and 2.4 are known as **small-signal equations**. For p-channel devices, μ_n, W_n, L_n, V_{tn}, and V_{GS} in the preceding equations are replaced with μ_p, W_p, L_p, $V_{tp,}$ and V_{SG}, respectively. Note that the preceding equations assume the substrate to be zero-biased, where $V_{sb} = 0$. Considerations with body effect, channel-length modulation, and process variations, etc. can be found in the references with in-depth discussions.

With small V_{DS}, a MOS transistor's I_D is linearly related to V_{DS}. As V_{DS} increases beyond a certain value, I_D will start to tap off as illustrated in Figure 2.4. This means that a MOS transistor is essentially a nonlinear device.

Figure 2.5 illustrates the n-channel conditions with respect to V_{DS}. When voltage applied on the gate terminal is greater than V_{tn}, channel current I_D starts to flow between the drain and source terminals, as depicted in Figure 2.5a. When $V_{DG} >= -V_{tn}$, channel pinch-off takes place at the drain end, as depicted in Figure 2.5b.

There are several sources of capacitance within and in the periphery of a MOS transistor. Figure 2.6 illustrates their existences and notations. These capacitors are often known as **parasitic capacitors**, because their presence is due to the physical construction of the MOS device.

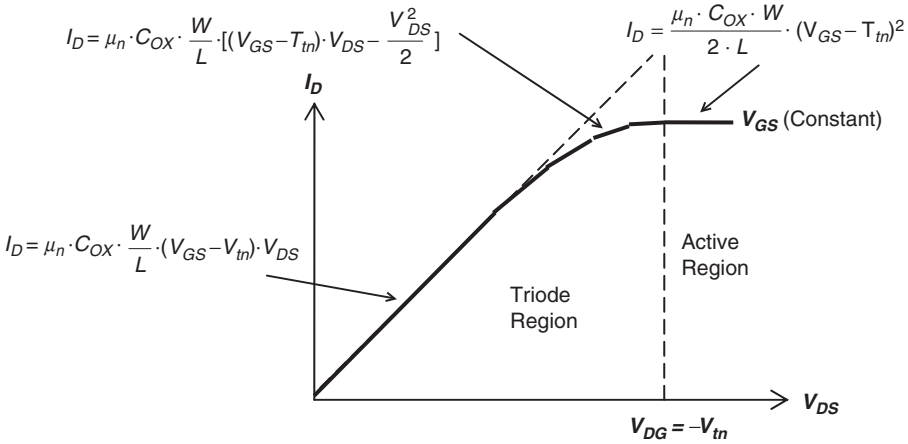


FIGURE 2.4

Nonlinear I_D *versus* V_{DS} relationship [Martin 2000].

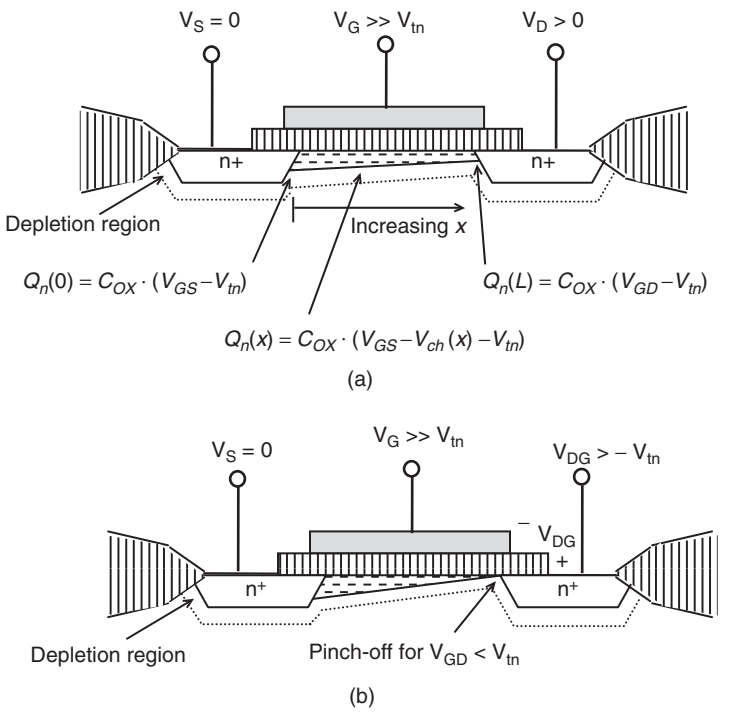


FIGURE 2.5

Illustration of n-channel conditions [Martin 2000]: (a) N-channel charge density. (b) N-channel pinch-off.

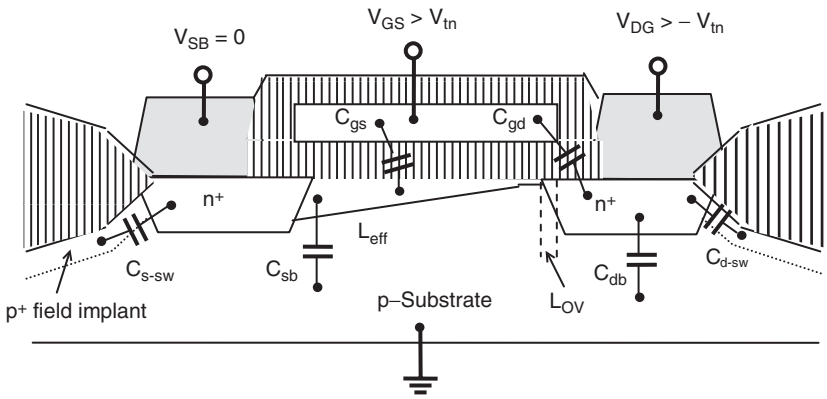


FIGURE 2.6

MOS device capacitance [Martin 2000].

It is worth noting that for IC engineering and manufacturing process control purposes, most transistors on the same chip are made with identical channel length. In addition, devices are often connected in parallel to form transistors having wider effective channels.

With nanometer technologies, process variations can affect the characteristics of individual transistors even on the same chips. We can no longer assume transistors on the same chip have the exact same threshold voltages. The ideal-case equations discussed in this section need to be adjusted to reflect process variation. We encourage readers to consult books on advanced CMOS modeling methods that take into account the effects of process variations.

2.2.2 **Transistor equivalency**

When a digital circuit uses many transistors, circuit analysis can get very complex and time-consuming. Transistor equivalency [Martin 2000] is a technique that simplifies larger circuits to smaller ones so that circuit analysis can be performed much more efficiently. The principles of transistor equivalency are illustrated in Figure 2.7. The first principle is **scaling**. When a MOS transistor's W and L are scaled by the same factor, as shown in Figure 2.7a, it has no effect on a first-order approximation. The second principle is called **parallel-connection equivalence**. When two MOS transistors T_1 and T_2 are connected in parallel, as shown in Figure 2.7b, the result is equivalent to a single transistor having the width equal to $W_1 + W_2$, with which

$$I_{D-eqv} = I_{D-1} + I_{D-2} = \frac{\mu \cdot C_{OX}}{2} \cdot \frac{W_1 + W_2}{L} \cdot (V_{GS} - V_t)^2 \qquad (2.5)$$

The third principle is called **serial-connection equivalence**, as depicted in Figure 2.7c, with which

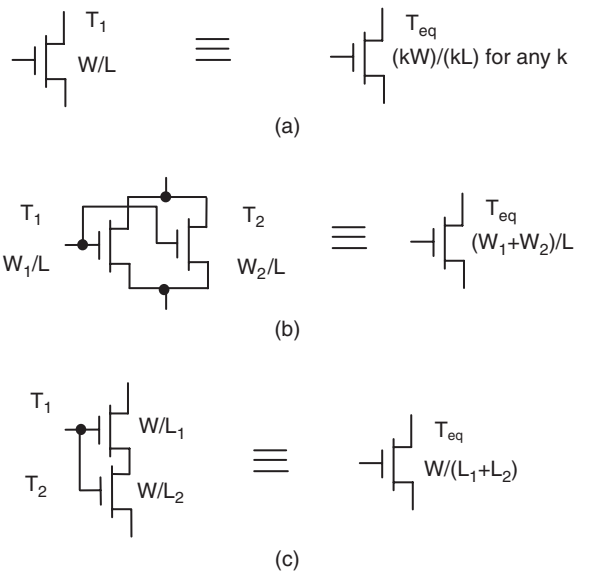


FIGURE 2.7

Illustration of transistor equivalency [Martin 2000]: (a) Scale equivalency. (b) Parallel-connection equivalency. (c) Serial-connection equivalency.

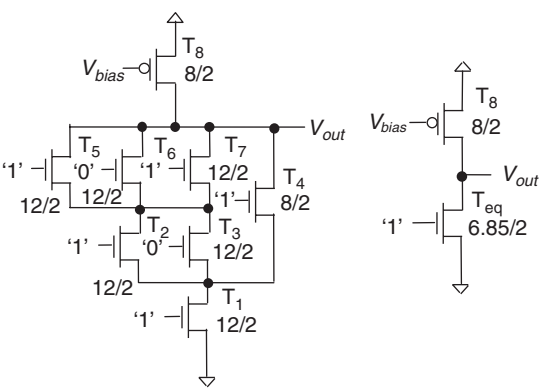


FIGURE 2.8

Application of transistor equivalency [Martin 2000].

$$I_{D-eqv} = I_{D-1} = I_{D-2} = \frac{\mu \cdot C_{OX}}{2} \cdot \frac{W}{L_1 + L_2} \cdot (V_{GS\_1} - V_t)^2 \tag{2.6}$$

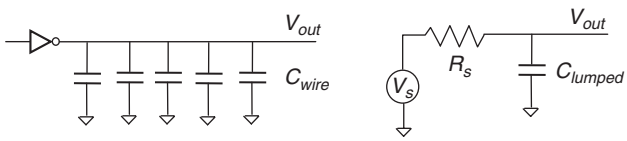
Consider the circuit shown in Figure 2.8. It uses the classic **pseudo-nMOS** technology, with which a single p-channel transistor (set by a constant biasing voltage, V_{bias}) is used as the load, whereas the inputs determine the switching

states of the n-channel transistors, which in turn determine the output of the circuit block. To apply transistor equivalency, the first step is to identify the n-channel transistors whose gate terminals are applied with "0" signals, because these transistors (T_3 and T_6, in this case) are set to the OFF state and can be ignored. Next, T_5 and T_7 are in parallel and are merged into a single one, T_5*, with W = 24/L = 2. Because T_5* and T_2 are in series, an equivalent transistor T_2* can be determined by first scaling T_2 to W = 24/L = 4 and then computing T_2* size as W = 24/L = 6. Repeat the same steps with T_4 followed by T_1. The resulting equivalent transistor, T_1*, is to have the size W = 6.857/L = 2. The resulting equivalent circuit is much easier to analyze than the original circuit with the given inputs.

2.2.3 Wire and interconnect

With CMOS technologies scaling down to the nanometer arena, wires that connect transistors to each other are becoming a dominant factor in almost all aspects of IC manufacturing, ranging from complexity and timing to silicon area and yield. Advanced CMOS technologies today provide 9 to 11 metal layers in interconnect space. Many ***application-specific integrated circuits*** (ASICs) require at least 7 metal layers to connect transistors.

For a typical single wire, the ***resistance-capacitance*** (RC) effects are distributed along its length, as illustrated in Figure 2.9a. However, the lumped RC model, as illustrated in Figure 2.9b, is often used for circuit analysis. Figure 2.10 illustrates the RC tree network of a source driving a number of output branches (*a.k.a.* fanouts).



(a) Distributed *C* model (b) Lumped *C* model

FIGURE 2.9

RC models for wire [Rabaey 2003].

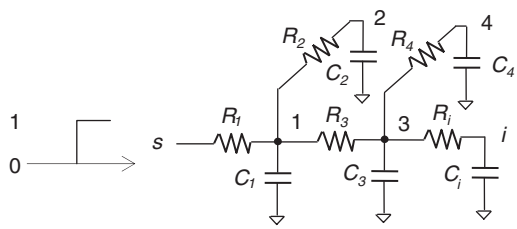


FIGURE 2.10

A tree-structured RC wire model [Rabaey 2003].

To calculate the RC effects between two nodes denoted as $\tau_{i,j}$ with i the source node and j the destination node, we have the following for the nodes in Figure 2.10:

$$\tau_{s,2} = C_1 \cdot R_1 + C_2 \cdot (R_1 + R_2) + (C_3 + C_4 + C_i) \cdot R_1$$
$$\tau_{s,4} = C_1 \cdot R_1 + C_2 \cdot R_1 + (C_3 + C_1) \cdot (R_1 + R_3) + C_4 \cdot (R_1 + R_3 + R_4)$$
$$\tau_{s,i} = C_1 \cdot R_1 + C_2 \cdot R_1 + (C_3 + C_4) \cdot (R_1 + R_3) + C_i \cdot (R_1 + R_3 + R_i)$$

As an exercise, readers are encouraged to figure out $\tau_{i,j}$ for other pairs of nodes.

In multilayer interconnect designs, wires placed in higher layers are usually wider and thicker than those in the lower layers, as illustrated in Figure 2.11, in which a six-metal layer hierarchy is depicted. This is to reduce resistance of long interconnects, because they are often placed in metal layers higher in the hierarchy. Lower metal layers are often reserved for shorter connections and for special purposes (such as distributing clocks). In addition, wires in higher layers are separated farther from each other to reduce coupling effects.

Coupling (inductive as well as capacitive) effects (*a.k.a.* **crosstalk**) between two or more parallel wires can affect signal integrity with unwanted circuit noise. **Coupling effects** also exist between wires on different layers. When long wires are placed in parallel next to each other, special care must be taken to reduce these effects.

Many of the IC routing technologies use two adjacent interconnect layers to complete one wiring. One layer would contain wires placed in North–South directions, and the other layer would contain wires placed in East–West directions. One advantage of this routing method is reduced interference between wires placed on adjacent layers. For this reason, wires on the two layers usually have the same width and thickness.

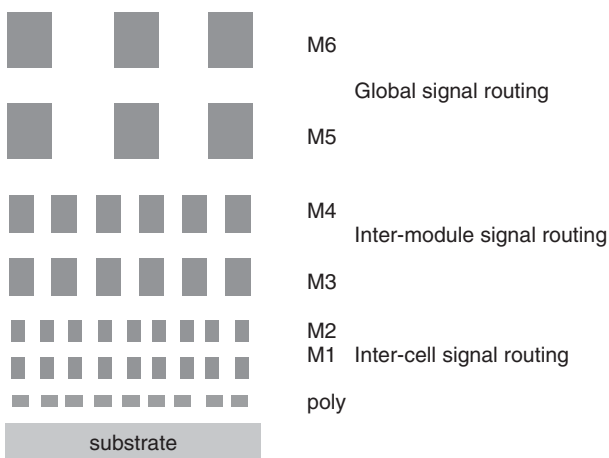


FIGURE 2.11

Multilayer interconnect hierarchy [Rabaey 2003].

2.2.4 Noise margin

Noise margin is a measure of design margins to ensure circuits functioning properly within specified conditions. Sources of noise include the operation environment, power supply, electric and magnetic fields, and radiation waves. On-chip transistor switching activity can also generate unwanted noise. To ensure that transistors switch properly under specified noisy conditions, circuits must be designed with specified **noise margins**.

Figure 2.12 illustrates noise margin and the terms, assuming that the signal generated by the driving device is wired to the input of the receiving device and that the wire is susceptible to noise. The minimum output voltage of the driving device for logic high, $V_{OH\ \min}$, must be greater than the minimum input voltage, $V_{IH\ \min}$, of the receiving device for logical high. Because of noise being induced on the wire, a logic high signal at the output of the driving device may arrive with lower voltage at the input of the receiving device. The noise margin, $NM_H = |V_{OH\ \min} - V_{IH\ \min}|$, for logical high is the range of tolerance for which a logical high signal can still be received correctly. The same can be said with noise margin, $NM_L = |V_{IL\ \max} - V_{OL\ \max}|$, for logical low, which specifies the range of tolerance for logical low signals on the wire. Smaller noise margins mean circuits are more sensitive to noise.

It is important to note that as CMOS technologies continue to advance, device feature size gets smaller, and channel length gets shorter. The miniaturization of transistors forces ever lower supply voltages, resulting in smaller noise margins. Table 2.1 shows the typical noise margin measurements with respect to technology advances.

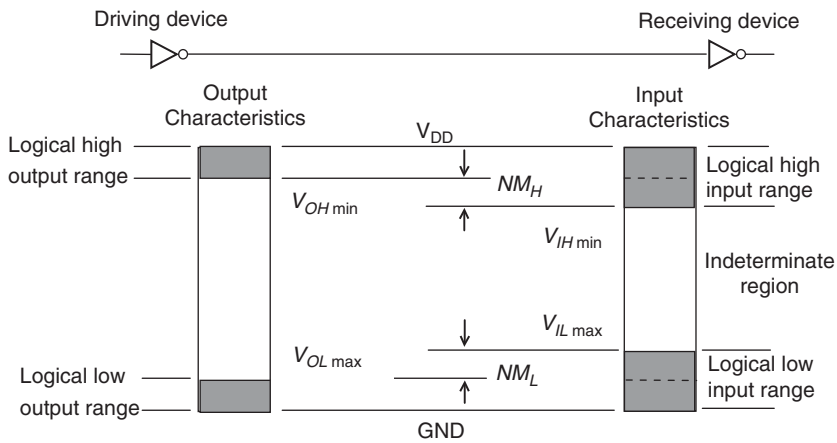


FIGURE 2.12

Noise margin and terms.

Table 2.1 Noise Margin Measures for Some Technologies [Wakerly 2001]

	Noise-Margin Measures					
Technology	V_{DD}	V_{OH}	V_{IH}	V_{TH}	V_{IL}	V_{OL}
5-V CMOS	5.0	4.44	3.5	2.5	1.5	0.5
5-V TTL	5.0	2.4	2.0	1.5	0.8	0.4
3.3-V LVTTL	3.3	2.4	2.0	1.5	0.8	0.4
2.5-V CMOS	2.5	2.0	1.7	1.2	0.7	0.4
1.8-V CMOS	1.8	1.45	1.2	0.9	0.65	0.45

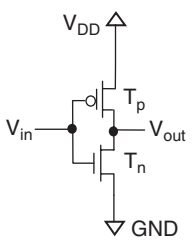
2.3 CMOS LOGIC

In this section we highlight some CMOS circuit design principles. We first review the classic CMOS inverter, with which the major measurements are discussed. The principles are carried over to the design of elementary logic gates and complex circuit blocks. Next, we discuss the design of latches and flip-flops, followed by discussion of some simple circuit optimization techniques.

2.3.1 CMOS inverter and analysis

The CMOS inverter consists of a pair of p-channel and n-channel transistors, as shown in Figure 2.13. Unlike pseudo-nMOS circuits, the p-channel transistor in this CMOS inverter is also a switching device, always in a complement switching state of the n-channel transistor, as shown in the truth table in Figure 2.13. Timing characteristics of this CMOS inverter include three measurements: t_r as the **rise time** at the output, t_f as the **fall time,** and t_p as the **propagation time** (*a.k.a.* **delay**) between an input transition and the output response. Figure 2.14 illustrates these measurements in graphic form.

Note that t_r and t_f are measured graphically by the pair of 10% and 90% change points on the output transition curves. In practice, however, the two intersecting



V_{in}	T_p	T_n	V_{out}
high	OFF	ON	low
low	ON	OFF	high

FIGURE 2.13

CMOS inverter and transistor state table.

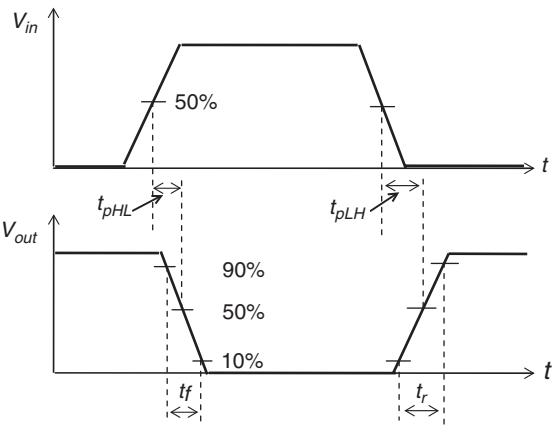


FIGURE 2.14

Illustrations of t_r, t_f, and t_p measurements [Rabaey 2003].

points on each transition curve by horizontally overlaying $V_{IH\ \min}$ and $V_{IL\ \max}$ are used. For $V_{DD} = 3.3V$, estimates of t_r and t_f can also be obtained as follows:

$$t_r = \frac{C_L}{I_{D-p}} \cdot \Delta V_{out} \approx \frac{2 \cdot C_L \cdot \Delta V_{out}}{\mu_p \cdot C_{OX} \cdot \frac{W_p}{L_p} \cdot \left(\frac{V_{DD}}{2} + V_{tp}\right)^2} \tag{2.7}$$

and

$$t_f = \frac{C_L}{I_{D-n}} \cdot \Delta V_{out} \approx \frac{2 \cdot C_L \cdot \Delta V_{out}}{\mu_n \cdot C_{OX} \cdot \frac{W_n}{L_n} \cdot \left(\frac{V_{DD}}{2} - V_{tn}\right)^2} \tag{2.8}$$

where C_L is the collective capacitance on the output of the CMOS inverter.

In practice, for process control and meeting engineering objectives (such as yield), both types of transistors are often manufactured with identical channel length. With this in mind and on the basis of Equations 2.7 and 2.8, making $t_r = t_f$ leads to

$$\left.\frac{W_p}{W_n}\right|_{t_r = t_f} = \frac{\mu_n \cdot (V_{DD} - V_{tn})}{\mu_p \cdot (V_{DD} + V_{tp})} \tag{2.9}$$

With most CMOS technologies this W_p/W_n ratio (for $t_r = t_f$) is between 1.5 and 3. Readers are encouraged to substitute data for specific technologies and verify.

Instead of $t_r = t_f$ being used, sometimes the criteria can be to minimize the average rise and fall time, where

$$t_{avg-r-f} = \frac{t_r + t_f}{2} \tag{2.10}$$

Substituting Equation 2.10 with Equations 2.7 and 2.8 and assuming $L_n = L_p = L$, we have

$$t_{avg\_r\_f} = C_L \cdot \Delta V_{out} \cdot \frac{L}{C_{OX}} \cdot \left(\frac{1}{\mu_p \cdot W_p \cdot \left(\frac{V_{DD}}{2} + V_{tp}\right)^2} + \frac{1}{\mu_n \cdot W_n \cdot \left(\frac{V_{DD}}{2} - V_{tn}\right)^2}\right) \tag{2.11}$$

Assuming that $C_L \approx C_{OX} \cdot L \cdot (W_n + W_p)$ and $|V_{tn}| \simeq |V_{tp}|$, the optimal W_p/W_n ratio is obtained by first rearranging Equation 2.11 to:

$$\begin{aligned} t_{avg\_r\_f} &\approx \frac{\Delta V_{out} \cdot L^2}{\mu_n \cdot \left(\frac{V_{DD}}{2} - V_{tn}\right)^2} \cdot \left(1 + \frac{\mu_n \cdot W_n}{\mu_p \cdot W_p}\right) \cdot \left(1 + \frac{W_p}{W_n}\right) \\ &= \frac{\Delta V_{out} \cdot L^2}{\mu_p \cdot \left(\frac{V_{DD}}{2} + V_{tp}\right)^2} \cdot \left(1 + \frac{\mu_n \cdot W_n}{\mu_p \cdot W_p}\right) \cdot \left(1 + \frac{W_p}{W_n}\right) \end{aligned} \tag{2.12}$$

and then differentiating Equation 2.12 with respect to W_p/W_n as:

$$\begin{aligned} \frac{\partial(t_{avg\_r\_f})}{\partial(W_p/W_n)} &= \frac{\Delta V_{out} \cdot L^2}{\mu_n \cdot \left(\frac{V_{DD}}{2} - V_{tn}\right)^2} \cdot \left[1 - \frac{\mu_n}{\mu_p} \cdot \left(\frac{W_p}{W_n}\right)^2\right] \\ &= \frac{\Delta V_{out} \cdot L^2}{\mu_p \cdot \left(\frac{V_{DD}}{2} + V_{tp}\right)^2} \cdot \left[1 - \frac{\mu_n}{\mu_p} \cdot \left(\frac{W_p}{W_n}\right)^2\right] \end{aligned} \tag{2.13}$$

and finally setting Equation 2.13 to zero. Therefore, we have:

$$\left.\frac{W_p}{W_n}\right|_{\min -\_t_{avg\_r\_f}} = \sqrt{\frac{\mu_n}{\mu_p}} \tag{2.14}$$

For many CMOS technologies, this W_p/W_n ratio (minimizing $t_{avg\_r\_f}$) is approximately 2. In practice, Equations 2.9 and 2.14 are often applied in sizing transistors.

Compared with a pseudo-nMOS inverter, this CMOS inverter consumes much less energy, because there is no direct current path between V_{DD} and the ground. Power dissipation of the CMOS inverter has three types: static, dynamic, and short-circuit. The **static power dissipation** is proportional to the leakage current when the inverter is not switching; the **dynamic power dissipation** is proportional to the switching frequency; and the **short-circuit power dissipation** is proportional to t_r and t_f.

Ideally, when the CMOS inverter is in either output high (T_p is ON and T_n is OFF in Figure 2.13) or output low (T_p is OFF and T_n is ON) state, there should be no current passing through the two transistors. However, in either state, a small current (*a.k.a.* **leakage current**) passes through the OFF-state transistor, hence, causing static power dissipation. The channel leakage currents can be obtained by calculating the channel resistance in the OFF state. The average static power dissipation is then:

$$P_{static\_avg} = V_{DD} \cdot \frac{I_{leak\_n} + I_{leak\_p}}{2} \tag{2.15}$$

Dynamic power dissipation is proportional to operating frequency, f_{clock}, which is the synchronization clock(s) in most digital circuits. Assuming V_{in} is a square wave signal running at f_{clock}, the average dynamic power dissipation is:

$$P_{dyn\_avg} = C_L \cdot V_{DD}^2 \cdot f_{clock} \tag{2.16}$$

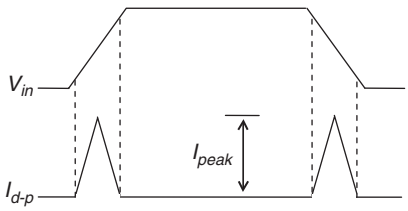


FIGURE 2.15

Illustration of direct-path current occurrences.

Short-circuit power dissipation is unique to CMOS circuits. It occurs while one of the two transistors is changing from the ON state to the OFF state and the other transistor from OFF to ON. During the transitions a direct-path current passes through both transistors. Figure 2.15 depicts the triangular $I_{d\text{-}p}$ waves.
The average short-circuit power dissipation is then:

$$P_{sc\_avg} = \mathrm{V_{DD}} \cdot I_{peak} \cdot \frac{t_r + t_f}{2} \cdot f_{clock} \tag{2.17}$$

and

$$I_{peak} = \frac{\mu_n \cdot C_{ox}}{2} \cdot \frac{W_n}{L_n} \cdot (V_{tb} - V_{tn})^2 \tag{2.18}$$

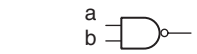
where V_{tb} is the threshold voltage of the CMOS inverter and V_{tn} is the threshold voltage of the n-channel transistor. The total average dynamic power dissipation is then:

$$P_{total\_dyn\_avg} = P_{dyn\_avg} + P_{sc\_avg} \tag{2.19}$$

2.3.2 Design of CMOS logic gates and circuit blocks

An **elementary CMOS logic gate** consists of an N-block and a P-block, each containing the number of corresponding channel transistors equal to the number of inputs of the gate. For example, with the 1-input CMOS inverter, the N-block contains one n-channel transistor and the P-block contains one p-channel transistor. Furthermore, the gate terminal of each n-channel transistor in the N-block is always connected to a corresponding p-channel transistor in the P-block. In addition, if two (or more) inputs are connected to the gate terminals of two n-channel transistors whose drain and source terminals are connected in series in the N-block, the same inputs are also connected to the gates terminals of two (or more) p-channel transistors whose drain and source terminals are connected in parallel.

Consider a 2-input (a and b) 1-output (c) NAND gate whose Boolean function is defined as $c = \overline{a \cdot b}$. Its symbol and truth table are shown in Figure 2.16,



INPUT		OUTPUT
a	b	c
Low	Low	High
Low	High	High
High	Low	High
High	High	Low

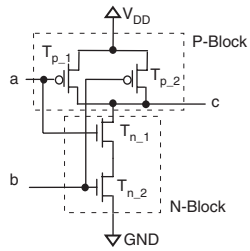
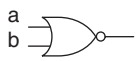


FIGURE 2.16

A NAND gate, its truth table, and a CMOS circuit implementation.



INPUT		OUTPUT
a	b	c
Low	Low	High
Low	High	Low
High	Low	Low
High	High	Low

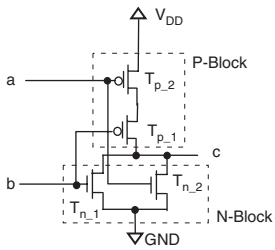


FIGURE 2.17

A NOR gate, its truth table, and a CMOS circuit implementation.

along with a typical CMOS circuit implementation. The AND operator (shown as ·) indicates that the two n-channel transistors controlled by the inputs must be placed next to each other in series and the two p-channel transistors controlled by the same inputs must be placed next to each other in parallel. When inputs a and b are both set to high, transistors T_{n_1} and T_{n_2} are turned ON such that output c is pulled down by means of discharge through the N-block, while both transistors in the P-block are OFF. In other input conditions at least one of the two transistors in the N-block is OFF and at least one of the two transistors in the P-block is ON, such that output c is being charged to high through the P-block.

Estimation of t_f is straightforward by identifying W_{n_eqv}, which comprises the width of both n-channel transistors. However, estimation of the rise time is somewhat complicated by the two p-channel transistors connected in parallel. Assuming that $W_{n\_1} = W_{n\_2}$ and $W_{p\_1} = W_{p\_2}$, which is often the case, then t_{r_min} is the rise time for both p-channel transistors to be turned ON and t_{r_max} is the rise time for only one of them to be turned ON, where $t_{r\_max} = 2\ t_{r\_min}$. It is often desired to make $t_f = t_{r\_max}$ in this and similar cases, for smaller W_{p_1} and W_{p_2}.

Figure 2.17 shows a typical CMOS implementation for a 2-input 1-output NOR gate whose Boolean function is defined as $c = \overline{a + b}$. When both inputs a and b are low, the output is driven to high by the P-block, because both

p-channel transistors are turned to ON and both n-channel transistors are turned to OFF. In other input conditions, at least one of the n-channel transistors is ON, pulling the output c down to low.

Similar to the analysis of the NAND gate, estimation of t_r is straightforward by identifying W_{p_eqv}, which comprises the width of both p-channel transistors. Because the two n-channel transistors are connected in parallel, the fall time comprises t_{f_min} (when both n-channel transistors are to be turned ON) and t_{f_max} (when only one of the two n-channel transistors is to be turned ON). Assuming that $W_{n\_1} = W_{n\_2}$, we have $t_{f\_max} = 2\ t_{f\_min}$. Oftentimes, it is desirable to also make $t_r = t_{f\_max}$ in this and similar cases.

To illustrate designing CMOS circuits implementing complex gates and random logic functions, as an example we use the carry bit circuit whose Boolean function is defined as $\overline{carry} = \overline{a \cdot b + (a + b) \cdot c}$ and a typical CMOS implementation is shown in Figure 2.18. In the N-block, transistors T_{n_3} and T_{n_5} implement $a \cdot b$, T_{n_1} and T_{n_2} for $a + b$, which is ANDed with c (implemented by T_{n_4}). Note that to implement the two ORs, T_{n_3} and T_{n_5} are placed in parallel alongside the other three n-channel transistors (for the first OR); T_{n_1} and T_{n_2} are placed in parallel with each other (for the second OR); T_{n_3} is placed in series with T_{n_5} to implement the first AND; and T_{n_4} is placed in series with T_{n_1} and T_{n_2} to implement the second AND.

Configuring the p-channel transistors in the P-block is to complement the configurations of the n-channel transistors. Here, T_{p_3} and T_{p_5} are placed in parallel with each other to complement T_{n_3} and T_{n_5}; T_{p_1} and T_{p_2} are placed in series to complement T_{n_1} and T_{n_2}; and T_{p_4} complements T_{n_4} and is placed in parallel with T_{p_1} and T_{p_2}, which are then placed in series with T_{p_3} and T_{p_5}.

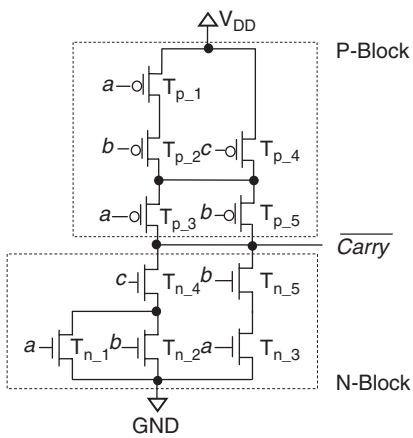


FIGURE 2.18

A CMOS implementation of a carry bit.

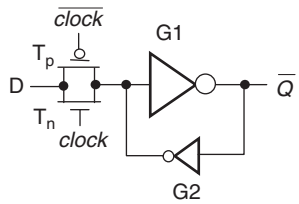


FIGURE 2.19

Implementation of a transmission-gate–based D latch.

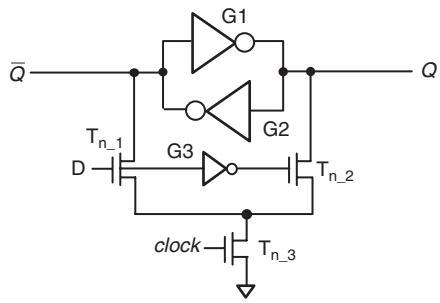


FIGURE 2.20

Implementation of an inverter-based D latch.

2.3.3 **Design of latches and flip-flops**

The simplest latch implementation uses two cross-coupled inverters and one **transmission gate**, as shown in Figure 2.19. The positive feedback allows the holding of a single bit of data at the output of G1 with its **collective load capacitance**. Transistors T_n and T_p are functioning together as a transmission gate. When the transmission gate is turned ON by the clock, the output bit $\bar{Q}$ is updated by the input D with $\bar{Q} = \bar{D}$. For this implementation to work reliably, the feedback inverter G2 must be significantly (approximately 10 times) smaller than the forward inverter G1. A smaller G2 will not interfere with input D to drive the G1 as desired.

Figure 2.20 shows an inverter-based D latch design with both Q and $\bar{Q}$ outputs. In this design, inverters G1 and G2 of identical sizes form the cross-coupled loop to hold a single bit of data. When the clock turns T_{n_3} to ON, input D will turn either T_{n_1} or T_{n_2} ON such that the outputs will be updated accordingly. When T_{n_3} is turned OFF, input D is disconnected from internal signals, and outputs Q and $\bar{Q}$ are driven by the **cross-coupled inverters** with the stored data. Note that G3 is a small inverter, because it only drives one transistor. By sizing the transistors properly, this inverter-based D latch can produce outputs Q and $\bar{Q}$ with similar timing characteristics. Figure 2.21 shows another inverter-based D latch implementation of two complementary outputs with the same timing measures—a characteristic important for **dual-rail processing**.

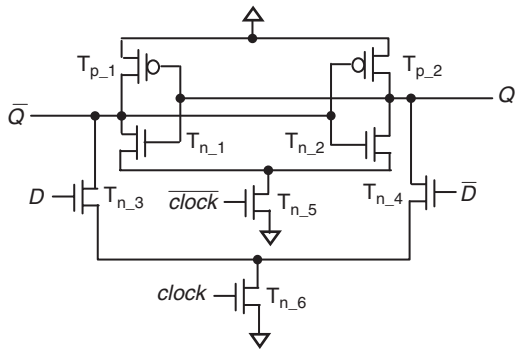


FIGURE 2.21

Implementation of a dual-rail inverter-based D latch.

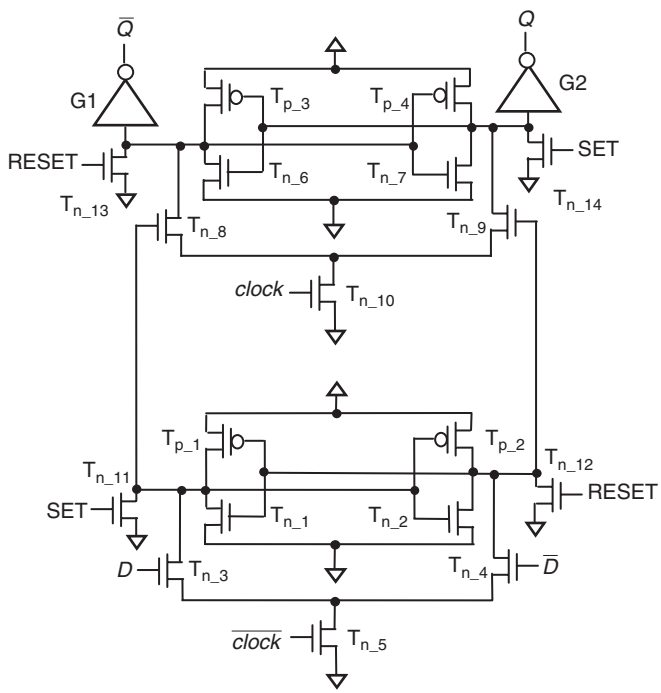


FIGURE 2.22

Implementation of a positive edge–triggered D flip-flop [Martin 2000].

A typical flip-flop contains two latches: one is called a master latch and the other is called a slave latch. The two latches work in complementary modes: when one latch is updating its content, the other is holding its outputs. Figure 2.22 shows a positive-edge-triggered dual-rail D flip-flop with asynchronous SET and RESET. Larger inverters G1 and G2 give greater driving capability. The SET and RESET functions are carried out in both the master and the slave latches.

2.3.4 **Optimization techniques for high performance**

In this section, we highlight several techniques for improving circuit performance. Other techniques that optimize circuits for low-power applications will be discussed in Section 2.6.

To improve circuit performance, it is often desirable to minimize the maximum number of transistors in series in the N-block and P-block. Consider the circuit shown in Figure 2.18. In the N-block, any path between the output and GND consists of two transistors. However, for the P-block there can be either two or three transistors between the output and V_{DD}. Carefully reviewing transistor configurations in the P-block, an equivalent implementation can be devised by rearranging the connections of the p-channel transistors as shown in Figure 2.23. This equivalent implementation has symmetric transistor configurations between the N-block and the P-block, hence improving performance.

Sometimes a small transistor is used to improve circuit performance. Figure 2.24 illustrates the concept of the use of a small **full-swing transistor** (*a.k.a.* keeper). As V_{out} goes low, T_p is turned ON, providing additional pulling of V_{in} to V_{DD}, which, in turn, speeds up V_{out} going low faster. When a CMOS logic block takes inputs from a pass-transistor logic block, the addition of this

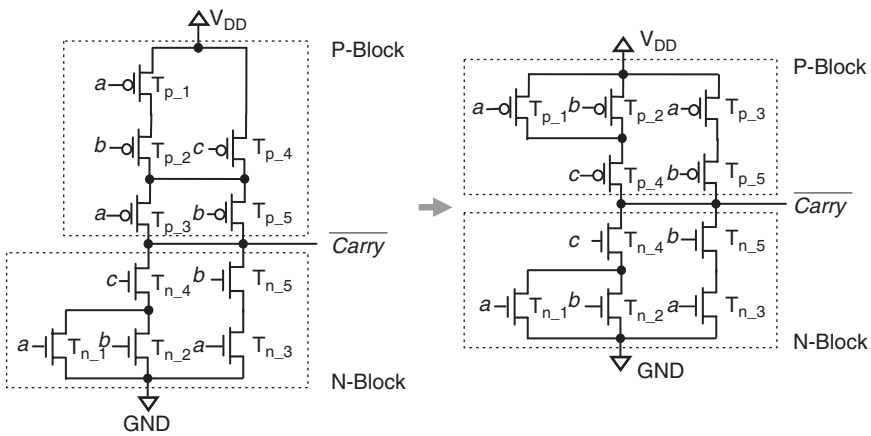


FIGURE 2.23

An optimized implementation of a carry bit.

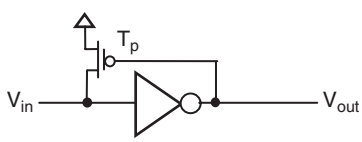


FIGURE 2.24

Application of a small full-swing transistor.

p-channel transistor eliminates the voltage drop because of the pass transistor. Note that the addition of T_p improves the t_f measure on V_{out}. Hence, it is a technique often used to balance circuit-timing measurements and optimize circuit implementations.

Because large digital systems often contain more than half a million latches in data path circuit structures and control logics, at times it becomes desirable to optimize their designs for a smaller area on silicon (*a.k.a.* footprint), as well as less power dissipation. Figure 2.25 shows a design known as an inverter-based three-state **dynamic latch**. T_{n_1} and T_{p_1} function as a traditional inverter. T_{n_2} and T_{p_2} control the periodical updating of the V_{out} node according to V_{in}. Capacitor C_{jp}, which is not explicitly included but rather is used to represent the junction and parasitic capacitance on the node, provides the single bit storage. This dynamic latch is approximately half the size of the transmission gate–based D latch shown in Figure 2.19 and approximately one fifth the size of the inverter-based D latch shown in Figure 2.20.

It should be pointed out that with the dynamic latch, as the data is stored on C_{jp}, the periodic updating (*a.k.a.* refresh) of V_{out} by *clock* must be performed before C_{jp} loses its charge through leakage to the substrate. Higher refresh rates mean higher power dissipation, which sometimes can be prohibitive. Meeting the clock frequency requirement with respect to C_{jp} and other design objectives can sometimes be challenging.

2.4 INTEGRATED CIRCUIT DESIGN TECHNIQUES

As modern digital systems demand more from circuit implementations, many new circuit technologies have emerged. These circuit technologies improve in one or more of the following areas: simplify implementation complexity, reduce silicon area, improve performance, and reduce power consumption. In this subsection, we highlight some of the techniques widely used in practice.

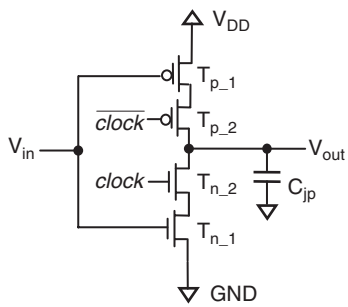


FIGURE 2.25

An inverter-based three-state dynamic latch.

2.4.1 **Transmission-gate/pass-transistor logic**

Transmission-gate/pass-transistor logic simplifies circuit implementations and yet does not require power supply to its circuit blocks. Consider a 2-to-1 multiplexer [Karim 2007]. Figure 2.26 compares a NAND gate implementation with a transmission-gate based implementation and a pass-transistor implementation.

The NAND-gate based implementation uses a total of 14 transistors, whereas the transmission-gate based and the pass-gate based implementations use 6 and 4 transistors, respectively. The NAND-gate based implementation incurs 2 gate delays between the data inputs and the output, whereas the transmission-gate based and the pass-transistor based implementations incur the channel resistance only.

One of the limiting factors with transmission-gate based and pass-transistor based implementations is the voltage drop when signals pass through them. Table 2.2 summarizes the transmission characteristics. Another is the higher internal capacitances in transmission-gate and pass-transistor configurations, because the junction capacitors are directly exposed to the signals passing through. Therefore, it is recommended that each transmission-gate based circuit block be followed with an active logic block, such as a CMOS inverter aided with a full-swing p-channel transistor (as shown in Figure 2.24).

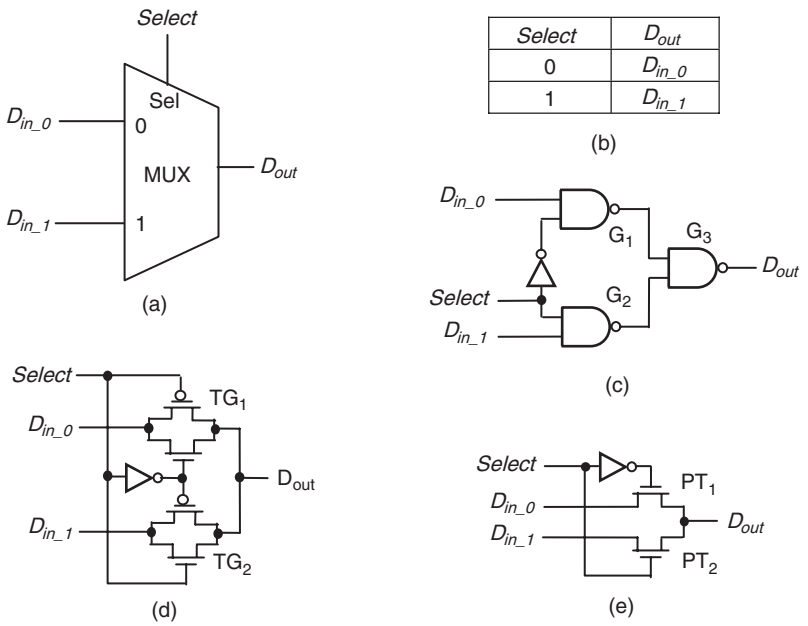


FIGURE 2.26

Comparison of 2-to-1 multiplexer implementations: (a) 2-to-1 MUX block symbol. (b) Truth table. (c) A NAND-gate-based implementation. (d) A transmission-gate-based implementation. (e) A pass-transistor-based implementaion.

Table 2.2 Measures of Transmission Characteristic [Wakerly 2001]

	Transmission Characteristic	
Device	**High**	**Low**
Transmission gate	Good	Good
N-channel pass transistor	Poor	Good
P-channel pass transistor	Good	Poor

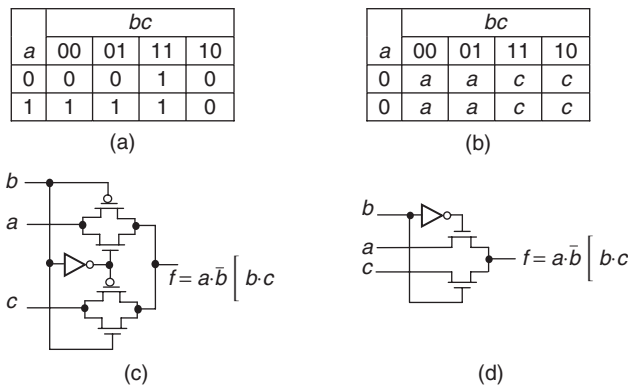


FIGURE 2.27

Comparison of 2-to-1 multiplexer implementations: (a) A normal Karnaugh map. (b) The modified Karnaugh map. (c) A transmission-gate-based design. (d) A pass-transistor-based design.

One of the key steps in the use of transmission gates and pass transistors for logic implementation is the identification of pass variable(s) to replace the 1's and 0's in normal Karnaugh maps. Instead of grouping 1's, as one would do in a normal Karnaugh map, variables are identified as pass variables or control variables and grouped accordingly. Pass variables are those to be connected to the data terminals of a multiplexer, whereas control variables are those to be connected to the select terminals. To illustrate this, consider a Boolean function $f(a, b, c) = a \cdot \bar{b} + b \cdot c$. Figure 2.27 shows the normal Karnaugh map (a) and its modified version (b) the use of pass variables, along with a transmission-gate based implementation (c) and a pass-transistor based implementation (d). After examining the normal Karnaugh map, one can conclude that when $b = 0$, the output f is determined by a; when $b = 1$, f is determined by c. This analysis results in the modified Karnaugh map, which indicates that b is the control variable, and a and c are the pass variables, resulting in the transmission-gate based and the pass-transistor based implementations shown in Figure 2.27. Readers are encouraged to try implementing other Boolean functions with this approach.

It should be noted that although transmission-gate based and pass-transistor based designs can reduce silicon area, placing a pass transistor on a normal signal path could lead to difficulty in testing, because a high-impedance state is introduced at the output of the pass transistor when the pass transistor is stuck at the OFF state.

2.4.2 **Differential CMOS logic**

Differential CMOS logic holds a unique place in dual-rail data processing circuits. This is because its two complementary outputs have identical timing characteristics. As illustrated in Figure 2.28, a differential CMOS circuit block consists of two symmetric left and right sub-blocks; each has one p-channel transistor in the P-block serving as the load device for the n-channel switching block below it. The two p-channel load devices are cross-coupled. The configurations of the n-channel transistors in the two sub–N-blocks follow the same AND-to-series OR-to-parallel constructions used with CMOS circuits. The symmetric circuit structures ensure identical timing characteristics at the two complementary outputs with respect to inputs.

Consider an XOR/XNOR combo block. Figure 2.29 compares three designs, an optimized CMOS NAND-based implementation (which is not for dual-rail), a differential CMOS logic implementation, and a hybrid of differential CMOS and pass-transistor implementation. With the CMOS NAND–based implementation shown in Figure 2.29b, the two complementary outputs have different delays. Hence, it is not suitable for dual-rail processing circuits. With the differential CMOS implementation shown in Figure 2.29c, the symmetric structures used by both output blocks ensure identical delay and, therefore, it is one of the desired circuit configurations for dual-rail processing. The implementation shown in Figure 2.29d simplifies the differential CMOS implementation by combining it with pass-transistor logic.

It should be noted that when complementary signals are not needed, the use of differential CMOS logic might result in a larger circuit footprint and more power consumption. Therefore, the circuit implementation must be chosen with respect to the requirements.

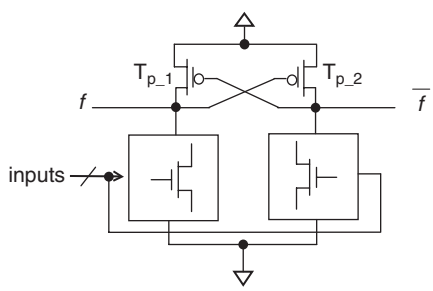
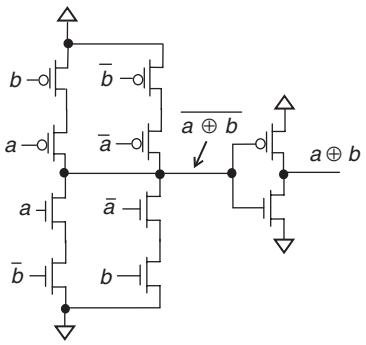


FIGURE 2.28

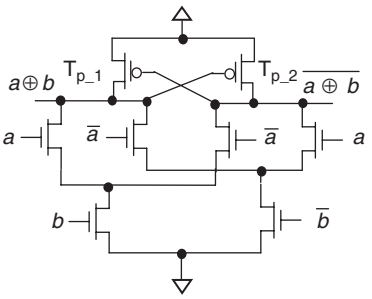
A generic diagram of a differential CMOS circuit block.

INPUTS		OUTPUTS	
a	*b*	*XOR*	*XNOR*
0	0	0	1
0	1	1	0
1	0	1	0
1	1	0	1

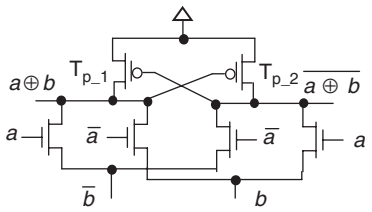
(a)



(b)



(c)



(d)

FIGURE 2.29

Comparison of implementations for XOR/XNOR: (a) Truth table for XOR/XNOR. (b) A differential CMOS implementation. (c) An optimized CMOS NAND-based implementation. (d) A hybrid implementation using differential CMOS and pass-transistor.

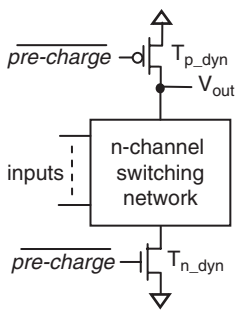


FIGURE 2.30

Generic structure of a dynamic pre-charge circuit block using n-channel switching transistors.

2.4.3 **Dynamic pre-charge logic**

Dynamic pre-charge logic has been widely used in high-performance microprocessors. Figure 2.30 illustrates the generic structure of a dynamic pre-charge circuit block, in which transistors T_{p_dyn} and T_{n_dyn} are **dynamic transistors**

and T_{p_dyn} is also known as the dynamic load. When the *pre-charge* signal is high, T_{p_dyn} is turned ON to charge the V_{out} node to high, while T_{n_dyn} is turned OFF to prevent currents going through the n-channel switching block to the ground. This period is called pre-charge phase, during which the output on V_{out} is ignored. This pre-charge phase is followed by an evaluation phase, during which T_{p_dyn} is turned OFF, T_{n_dyn} is turned ON, and V_{out} is determined by the n-channel switching network controlled by the inputs. If the inputs are evaluated for V_{out} to go low, the pre-charged voltage on V_{out} is discharged through the n-channel switching network, because it has at least one path connecting V_{out} to ground. Otherwise, V_{out} remains floating at the pre-charged high value.

Transistor configurations in the n-channel switching network follow the same design steps as those used for classic CMOS circuits. Figure 2.31 shows the NAND and NOR blocks using dynamic pre-charge logic.

Similarly, instead of using an n-channel switching network, dynamic pre-charge circuits can use p-channel switching transistors. A generic structure of dynamic pre-charge logic by use of a p-channel switching network is shown in Figure 2.32. During the pre-charge phases, T_{n_dyn} is turned ON and T_{p_dyn} is turned OFF, and V_{out} is discharged to low. During the evaluation phases, T_{n_dyn} is turned OFF and T_{p_dyn} is turned ON, and V_{out} is determined by the configurations of p-channel transistors in the p-channel switching network. If inputs are evaluated for V_{out} to go high, the output node gets charged from V_{DD} through at least one path in the p-channel switching network that connects V_{out} with V_{DD}. Otherwise, V_{out} remains low. Figure 2.33 shows the implementations for a 2-input NAND and 2-input NOR gate using p-channel switching transistors.

2.4.4 **Domino logic**

Cascading dynamic pre-charge logic blocks one after another may result in erroneous outputs because of a phenomenon known as **partial discharge**, as

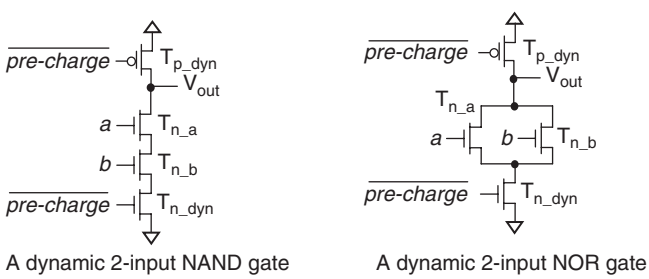


FIGURE 2.31

Dynamic 2-input NAND and NOR implementations using n-channel switching transistors gate.

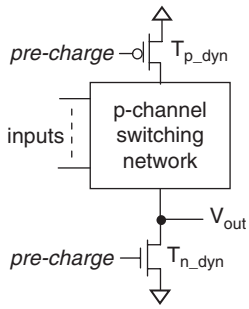


FIGURE 2.32

Generic structure of a dynamic pre-charge circuit block using p-channel switching transistors.

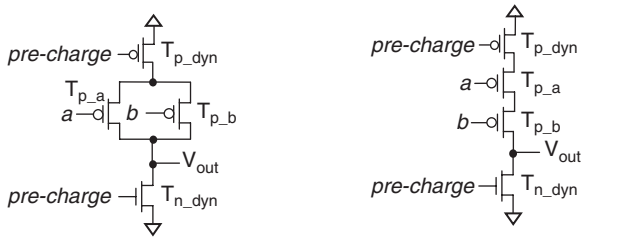


FIGURE 2.33

Dynamic 2-input NAND and NOR gate implementations using p-channel switching transistors.

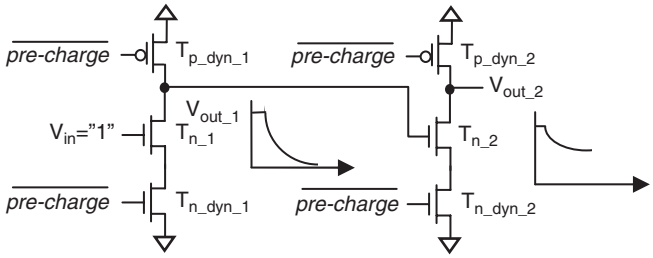


FIGURE 2.34

Partial discharge in cascaded dynamic pre-charge inverters.

illustrated in Figure 2.34 with respect to $V_{in} = 1$. First, both outputs of the two inverters will be pre-charged to high. Next, V_{out_1} is to be discharged to low. Ideally, V_{out_2} would remain high, because the input to the second inverter is going low. However, because T_{n_2} is initially in the ON state right after the evaluation

phase begins, V_{out_2} may be partially discharged, potentially resulting in an erroneous output. (Readers are encouraged to analyze cascaded dynamic inverters by use of p-channel switching transistors.) To avoid this partial discharge problem in practice, a dynamic pre-charge block is often followed by a CMOS inverter, and the resulting circuit structure is known as **Domino CMOS logic** whose generic circuit structure is illustrated in Figure 2.35.

To demonstrate the applications of Domino logic, consider a 4-bit comparator. The truth table for a single-bit slice comparator is shown in Table 2.3, and the Boolean function is $f(C_{in}, A, B) = A \cdot \overline{B} + A \cdot C_{in} + \overline{B} \cdot C_{in} = A \cdot \overline{B} + (A + \overline{B}) \cdot C_{in}$. By use of Domino logic with n-channel switching transistors, the single-bit comparator circuit implementation is shown in Figure 2.36, along with the 4-bit block diagram.

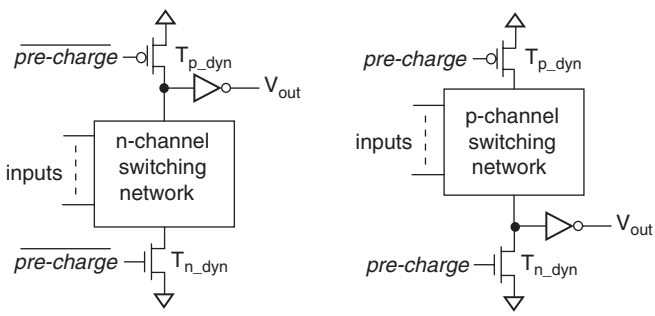


FIGURE 2.35

Generic structure of a Domino CMOS logic circuit block.

Table 2.3 Single-Bit Comparator

Inputs			**Output**
C_{in}	A	B	$A > B$
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

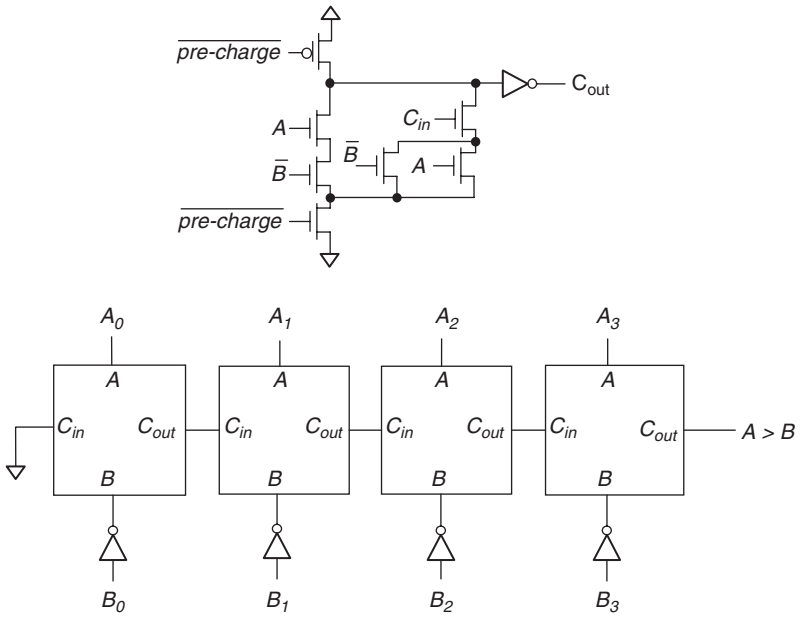


FIGURE 2.36

A 4-bit comparator implementation by use of Domino logic.

It should be pointed out that because transistor T_{p_dyn} acts as a dynamic load, the outputs of dynamic precharge logic and Domino logic will leak away over time and thus may not be valid in certain situations where clocking is halted. For example, when diagnosis of digital circuits is performed, it is often necessary for engineers to apply a certain number of clock cycles to a circuit, stop, and then probe selected signals to take necessary measurements. These and similar operations may not be possible with dynamic pre-charge and Domino logics, because they require constant pre-charge and evaluation cycles.

To overcome this shortcoming, a small (often of minimum size) static load p-channel transistor (*a.k.a.* **keeper**) is added alongside the dynamic load, as illustrated in Figure 2.37. This small keeper transistor provides just enough current to overcome the leakage current during probing, in the case with dynamic pre-charge logic, and it also improves the high-to-low transition at V_{out}.

For dynamic circuit blocks implementing complex logic functions, the n-channel switching network often contains many stacked transistors, which may cause erroneous outputs during the evaluation phases. The phenomenon is known as **charge sharing**, which is illustrated in Figure 2.38. During an evaluation phase, transistors *A*, *B*, and *E* are OFF and transistor D is ON, and the charge on C_1 is now shared with C_2, which is much bigger than C_1. This would cause the voltage at the input of the inverter to drop, which may lead to an erroneous V_{out}. To prevent this charge-sharing problem, selected internal nodes in

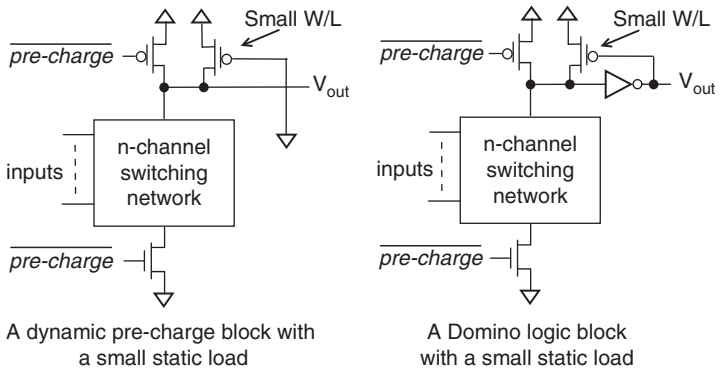


FIGURE 2.37

Illustration of dynamic circuit blocks with static load.

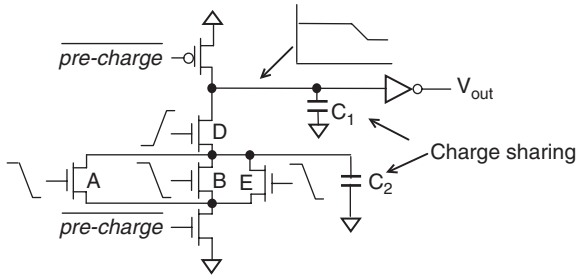


FIGURE 2.38

Charge sharing in a dynamic CMOS circuit.

the switching network can be pre-charged as well. This is illustrated in the implementation of a multi-output dynamic circuit block shown in Figure 2.39. No explicit dynamic transistor is placed at internal nodes where pre-charge is guaranteed. Readers are encouraged to identify these internal nodes as an exercise.

2.4.5 **No-race logic**

One of the limitations with Domino logic is the insertion of an inverter at each block's output. When Domino logic circuit blocks are cascaded, the added inverters can result in excessive delay. One way to reduce such delay is alternating between n-channel pre-charge blocks and p-channel pre-charge blocks, a technique known as **NORA** [Martin 2000] (for **no-race logic**), as illustrated in Figure 2.40, when dynamic circuit blocks are cascaded one after another.

A **dynamic latch** (*a.k.a.* clocked latch) has also been used in the place of the inverter in a Domino logic circuit block. During a pre-charge phase, the dynamic latch appears as high impedance. During an evaluation phase,

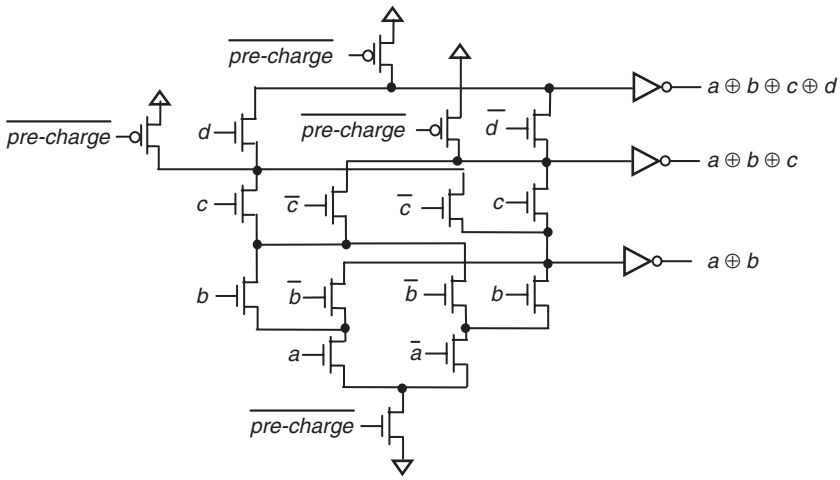


FIGURE 2.39

Precharge of selected internal nodes in a multi-output Domino logic circuit block.

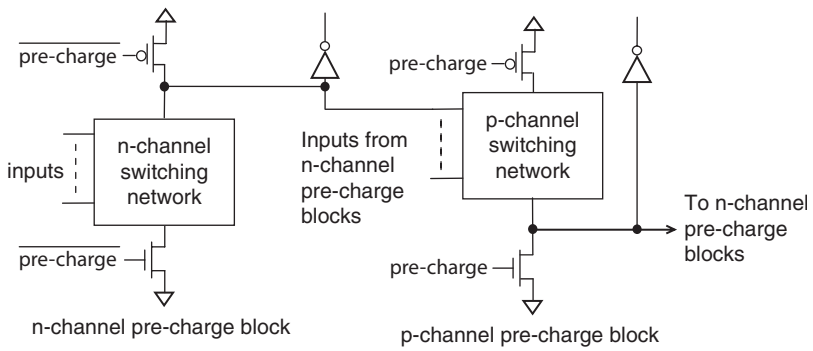


FIGURE 2.40

Altering n-channel pre-charge and p-channel pre-charge blocks.

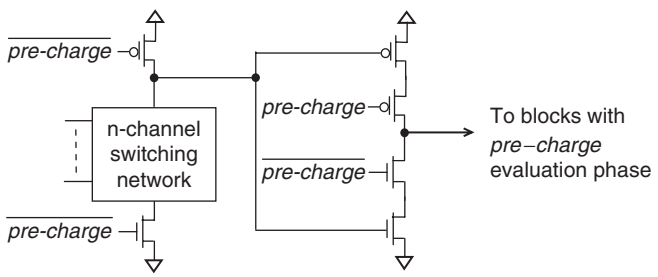


FIGURE 2.41

A dynamic circuit block with a dynamic latch output buffer.

the dynamic latch samples the output of the dynamic block and stores its output during the next pre-charge phase. The dynamic circuit block and the latch are pre-charged and evaluated in opposite phases, therefore, eliminating the partial discharge problem.

A circuit structure combining the preceding two approaches is known as *No-Race logic*, as illustrated in Figure 2.42 with two stages. The first is the $\overline{\textit{pre-charge}}$ evaluation stage because its circuit blocks are evaluated in that phase. This stage consists of an n-channel Domino block, which is followed by a p-channel Domino block, with the output being clocked by a dynamic latch. Outputs of the two Domino logic circuit blocks can feed other circuit blocks as indicated, without being latched. In the second stage, switching networks are evaluated in the *pre-charge* phase. Hence, this stage is called the

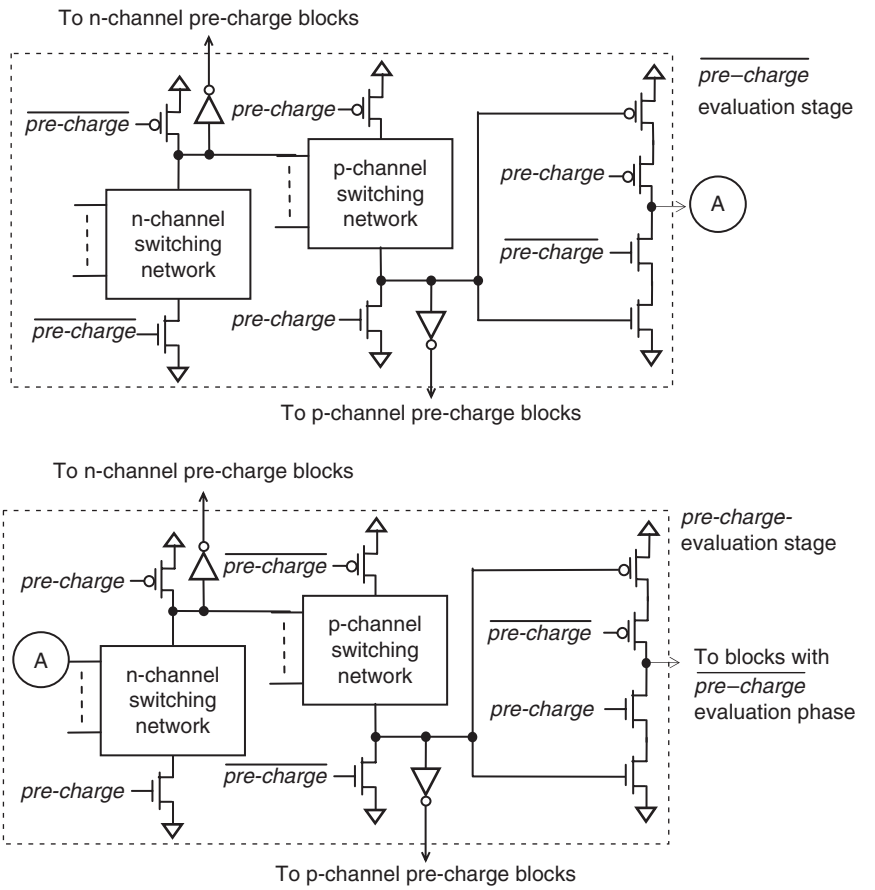


FIGURE 2.42

Circuit structure of No-Race logic.

pre-charge-evaluation stage. It consists of the same circuit components and structure as the first stage, except that the dynamic control signals are replaced with the complemented version. This two-stage section can be repeated several times to form highly efficient pipeline structures.

Note that the circuit blocks in the two-stage structure illustrated in Figure 2.42 use dynamic loads. When static loads are used, there are constraints on the number of inversions to guarantee race-free operation in the presence of clock skews. Techniques such as **reverse clock distribution** and **local clock generation** that use differential circuits are also used in practice to ensure race-free operation in high-performance CMOS circuits. For the analysis and design principles, readers are encouraged to explore further with the references listed at the end of this section.

2.4.6 Single-phase logic

As described and illustrated in the previous subsections on dynamic CMOS circuit implementations, both *pre-charge* and $\overline{\textit{pre-charge}}$ phases are used. Techniques that use only one phase are known as **single-phase logic**, which simplifies dynamic implementations. Figure 2.43 illustrates the generic diagram of two basic single-phase logic components, with one that uses an n-channel switching network and the other that uses p-channel switching network. Note

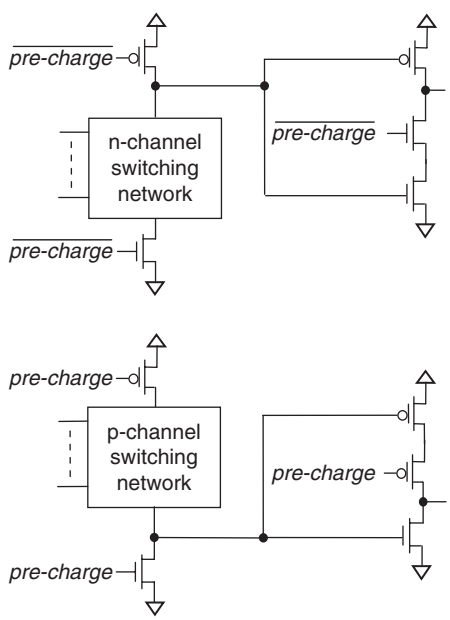


FIGURE 2.43

Generic diagram of single-phase logic blocks.

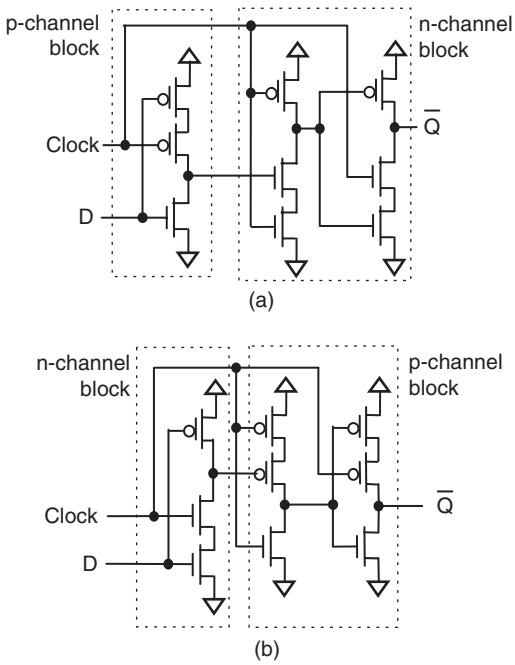


FIGURE 2.44

Single-phase edge-trigged dynamic D flip-flops: (a) Positive edge-triggered dynamic D flip-flop. (b) Negative edge-triggered dynamic D flip-flop.

that each dynamic circuit block uses one phase of the *pre-charge* signal. Figure 2.44 shows two single-phase edge-trigged dynamic D flip-flops. Readers are encouraged to analyze the way that these two dynamic flip-flops work. Single-phase logic can simplify the clock distribution that can be very complex in many large digital systems in which high-performance dynamic circuits are used.

2.5 CMOS PHYSICAL DESIGN

Once transistor schematics are ready, the next design step is to translate the circuit schematic designs into the device and wire placements on silicon. This design step is known as **physical design**, which produces silicon blueprints specifying the exact size and location of each transistor, wire, contact, and other components before manufacturing masks are generated.

Circuit simulation incorporating physical design specifics can more accurately mimic the real chip behavior than schematic-based circuit simulation. This is because at the circuit schematic level, oftentimes the exact length of each wire is not known yet. Therefore, circuit designs with small design margins are often simulated again with physical design data to further ensure that design metrics are satisfied.

In this section, we highlight some basic concepts and practices in physical design. For more in-depth study, readers are encouraged to explore the references further. To help with visualizing layout designs, the **Mead-Conway color-coordination** is often used to differentiate CMOS structures [Weste 1994]. Table 2.4 shows the color representation for the n-well CMOS process. When color display is not available, varying fill-in patterns and shades are used.

2.5.1 Layout design rules

Layout design rules specify geometric constraints with respect to physical constructs. These layout design rules are intended to ensure that designs can be properly manufactured through the manufacturing processes and satisfy all engineering metrics. Because layout design rules are technology and process specific, care must be taken to ensure that only certified layout design rules of the intended technology and processes are used.

Layout design rules are defined in terms of **feature sizes**, **separations,** and **overlaps**. Feature size defines the dimensions of constructs, such as the channel length and the width of wires. Separation defines the distance between two constructs on the same layer. Overlap defines the necessary overlap of two constructs on adjacent layers in a physical construction, such as a contact connecting a Poly wire with a Metal 1 wire, in which the Metal 1 wire must overlap with the Poly wire below. Table 2.5 lists two typical sets of CMOS layout design rules for an n-well–based process. One is called the **λ-Rule set** and the other is called the **μ-Rule set**. The λ-Rule set is scalable with λ (which is typically twice the channel feature size), therefore, giving designs much flexibility in choosing manufacturing facilities and stability in dealing with multiple manufacturing lines and vendors. The μ-Rule set specifies the exact feature

Table 2.4 N-Well CMOS Process Color-Layer Representation [Weste 1994]

Layer	Color	Symbolic
N-well	Brown	
Thin-oxide	Green	n-channel transistor
Poly	Red	Poly-silicon
p^+	Yellow	p-channel transistor
Contact-cut, via	Black	Contact
Metal 1	Blue	Metal 1
Metal 2	Tan	Metal 2
Metal 3	Gray	Metal 3
Metal 4	Purple	Metal 4

Table 2.5 CMOS Layout Design Rules [Weste 1994]

	λ-Rule	**μ-Rule**
A. N-well layer		
A.1 Minimum size	10λ	2μ
A.2 Minimum spacing (well at same potential)	6λ	2μ
A.3 Minimum spacing (well at different potential)	8λ	2μ
B. Active Area		
B.1 Minimum size	3λ	1μ
B.2 Minimum spacing	3λ	1μ
B.3 N-well overlap of p^+	5λ	1μ
B.4 N-well overlap of n^+	3λ	1μ
B.5 N-well space to n^+	5λ	5μ
B.6 N-well space to p^+	3λ	3μ
C. Poly		
C.1 Minimum size	2λ	1μ
C.2 Minimum spacing	2λ	1μ
C.3 Spacing to Active	1λ	0.5μ
C.4 Gate Extension	2λ	1μ
D. p^+/n^+		
D.1 Minimum overlap of Active	2λ	1μ
D.2 Minimum size	7λ	3μ
D.3 Minimum overlap of Active in substrate contact	1λ	2μ
D.4 Spacing of p^+/n^+ *to* n^+/p^+ gate	3λ	1.5μ
E. Contact		
E.1 Minimum size	2λ	0.75μ
E.2 Minimum space on Poly	2λ	1μ
E.3 Minimum space on Active	2λ	0.75μ
E.4 Minimum overlap of Active	2λ	0.5μ
E.5 Minimum overlap of Poly	2λ	0.5μ
E.6 Minimum overlap of Metal 1	1λ	0.5μ

continued

Table 2.5 CMOS Layout Design Rules [Weste 1994]—*cont.*		
E.7 Minimum space to Gate	2λ	1μ
F. Metal 1		
F.1 Minimum size	3λ	1μ
F.2 Minimum spacing	3λ	1μ
G. Via		
G.1 Minimum size	2λ	0.75μ
G.2 Minimum spacing	3λ	1.5μ
G.3 Minimum Metal 1 overlap	1λ	0.5μ
G.4 Minimum Metal 2 overlap	1λ	0.5μ
H. Metal 2		
H.1 Minimum size	3λ	1μ
H.2 Minimum spacing	4λ	1μ
I. Via 2		
I.1 Minimum size	2λ	1μ
I.2 Minimum spacing	3λ	1.5μ
I.3 Minimum Metal 2 overlap	2λ	1μ
I.4 Minimum Metal 3 overlap	3λ	1.5μ
J. Metal 3		
J.1 Minimum size	8λ	4μ
J.2 Minimum spacing	5λ	2.5μ
J.3 Minimum Metal 2 overlap	2λ	1μ
J.4 Minimum Metal 3 overlap	2λ	1μ
K. Passivation		
K.1 Minimum opening		100μ
K.2 Minimum spacing		150μ

sizes, required separations, and overlaps for a targeted line of technology and processes. It is often used for high-volume designs.

Entries in Table 2.5 are mostly self-explanatory. For example, Rule A.1 specifies that, for the intended n-well technology, the dimensions of the n-well must be at least $10\lambda \times 10\lambda$ in a layout design following the λ-Rule set and $2\mu \times 2\mu$

following the μ-Rule set. Rule A.2 specifies that the minimum space between two separate n-wells of the same potential must be 6λ and 2μ, respectively. Rule C.1 specifies that a Poly section must be 2λ wide with λ-Rule and 1μ with μ-Rule. Rule C.2 specifies that there must be at least 2λ (or 1μ) separation between two neighboring Poly sections. As readers may observe in Table 2.5, layout designs following the λ-Rule set almost always end up occupying more silicon space than those following the μ-Rule set. This is because the λ-Rule set incorporates built-in scalability, whereas the μ-Rule does not have this flexibility (therefore, it can be optimized for minimum use of the silicon area). Figures 2.45 and 2.46 illustrate graphically the layout design rules in Color and Black/White, respectively.

2.5.2 **Stick diagram**

Stick diagrams are useful tools for planning custom physical layout designs of complex circuit blocks. In a stick diagram, transistors are represented by colored sticks, contacts are represented by black dots, and wires are represented by lines; all are placed on a square-grid background. Transistor representations in a stick diagram are the same regardless of their size. Figure 2.47 illustrates two stick diagrams of a CMOS inverter, illustrating that different transistor placement orientations result in layouts with different aspect ratios.

One of the applications of a stick diagram is to investigate the best placement of transistors, including their orientations and relative positions. This is an important step in designing layouts of complex circuit blocks, because transistor placements can affect wiring complexity and many circuit performance characteristics. The common objectives used in devising stick diagrams are minimizing the overall block area and the use of wires. Other objectives can be proper alignment of input and output signals, such that when a block is to be cascaded in series, the layout block can be repeated without much reconnection. Oftentimes, layout design engineers can find themselves in a position in which minimizing block area and the use of wires cannot be achieved at the same time, and hence a tradeoff must be made to proceed. The simplicity of stick diagrams gives layout design engineers a "quick-and-dirty" approach to investigate the potential impacts to aid in making layout design decisions.

Another application of stick diagrams is for estimating the block layout dimensions. In this case, the background grid X and Y dimensions are indexed. With a given layout stick diagram along with the set of layout design rules, sizes of constructs on the X and Y axis are added up to determine the total length on that index. For example: X(3) for the stick diagram in Figure 2.47a passes through the width of the GND wire and the source contact of the n-channel transistor, the n-channel length, the n-channel transistor drain terminal contact, the separation space of the terminal contacts, the p-channel drain terminal contact, the p-channel length, the p-channel source terminal contact, and the V_{DD} wire; X(8) for stick diagram in Figure 2.47b intersects with the GND wire,

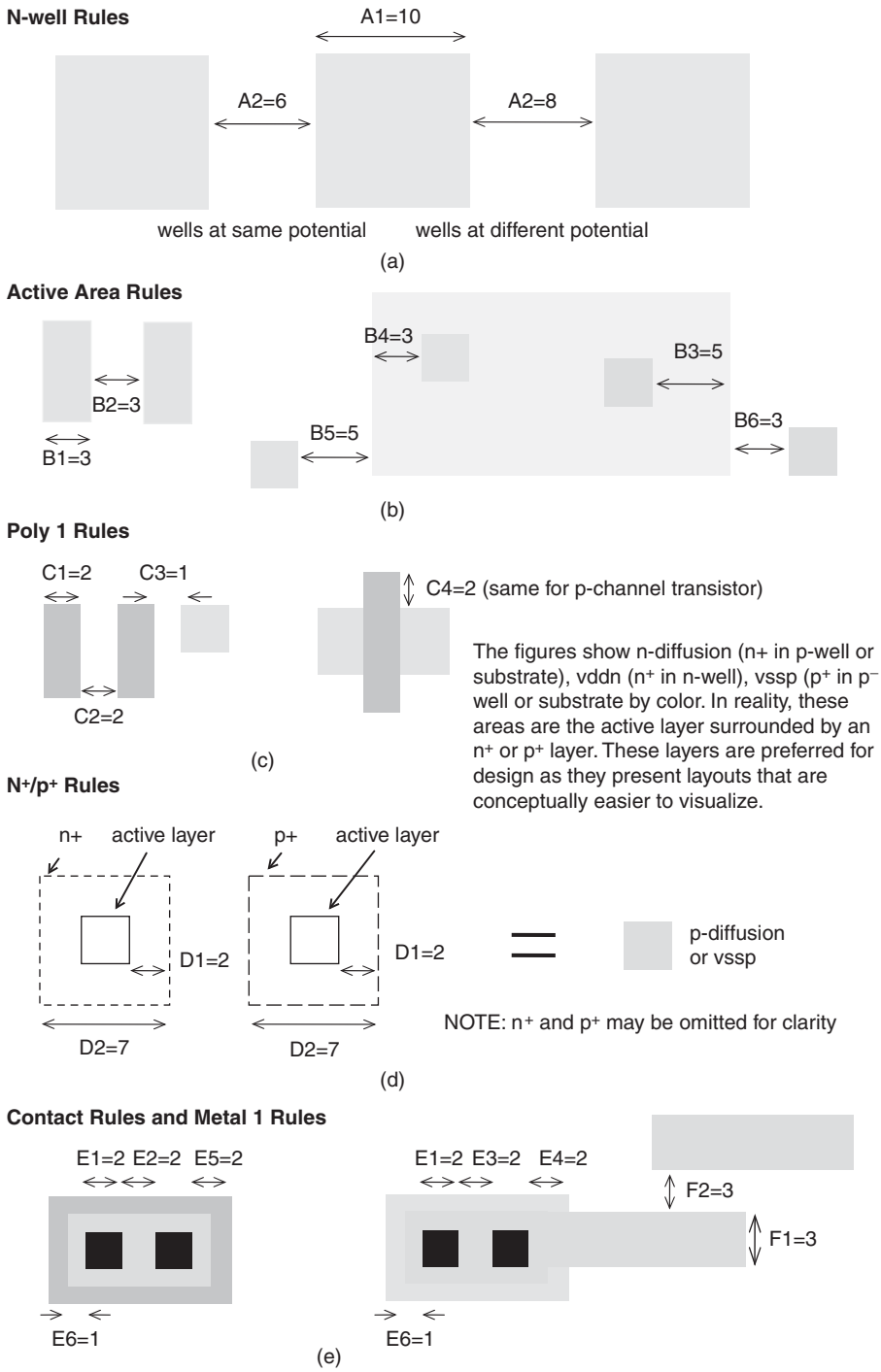


FIGURE 2.45

Continued

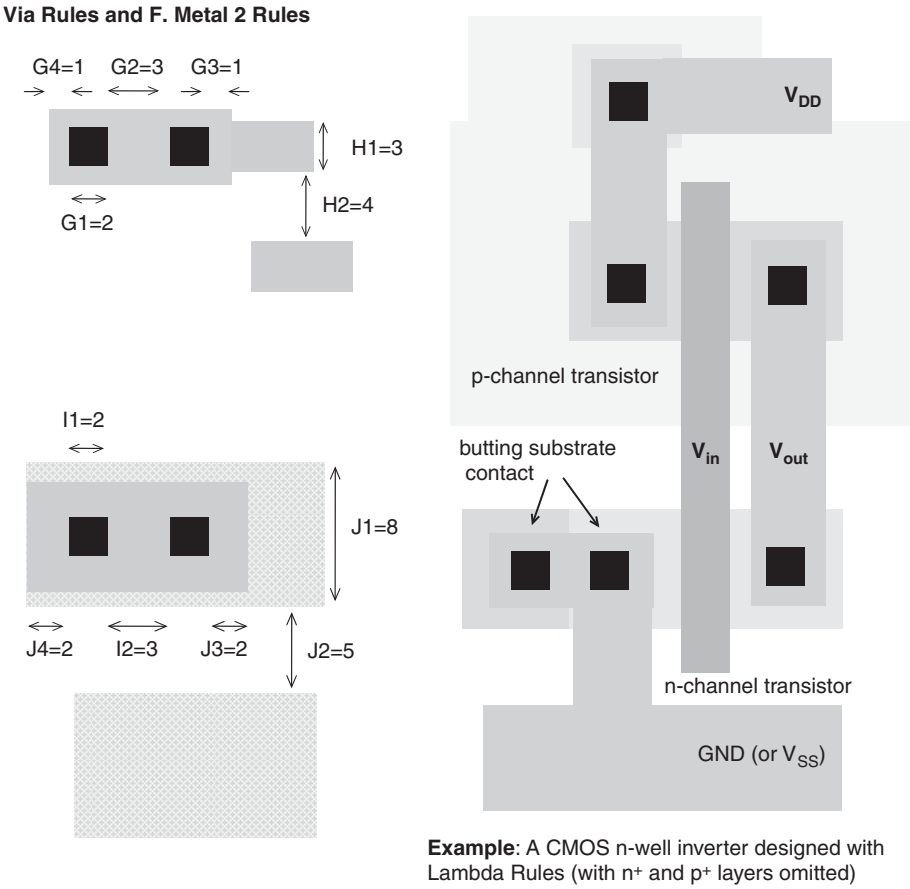


FIGURE 2.45

Illustration of layout rules and color designations [Weste 1994].

the source terminal contact of the n-channel transistor, spacing between M1 and the contact, the M1 wire, the source contact of the p-channel transistor, and the V_{DD} wire; X(9) goes through the GND wire, the n-channel gate extension, the width of the n-channel transistor, spacing between M1-Poly contact and the n-channel, the M1-Poly contact, spacing between p-channel and M1-Poly contact, the width of the p-channel transistor, the p-channel gate extension, the width of the V_{DD} wire. By use of the λ-Rule, Table 2.6 lists the estimates on the X and Y index for Figures 2.47a and 2.47b layouts, with the assumption that the transistors have an identical channel width of 2λ.

Because a custom physical layout design often requires several iterations of floorplanning, placement, and routing, estimates of block dimensions on the basis of stick diagrams can help to reduce the number of iterations, hence, improving the efficiency of design activities. Although in recent years, CAD

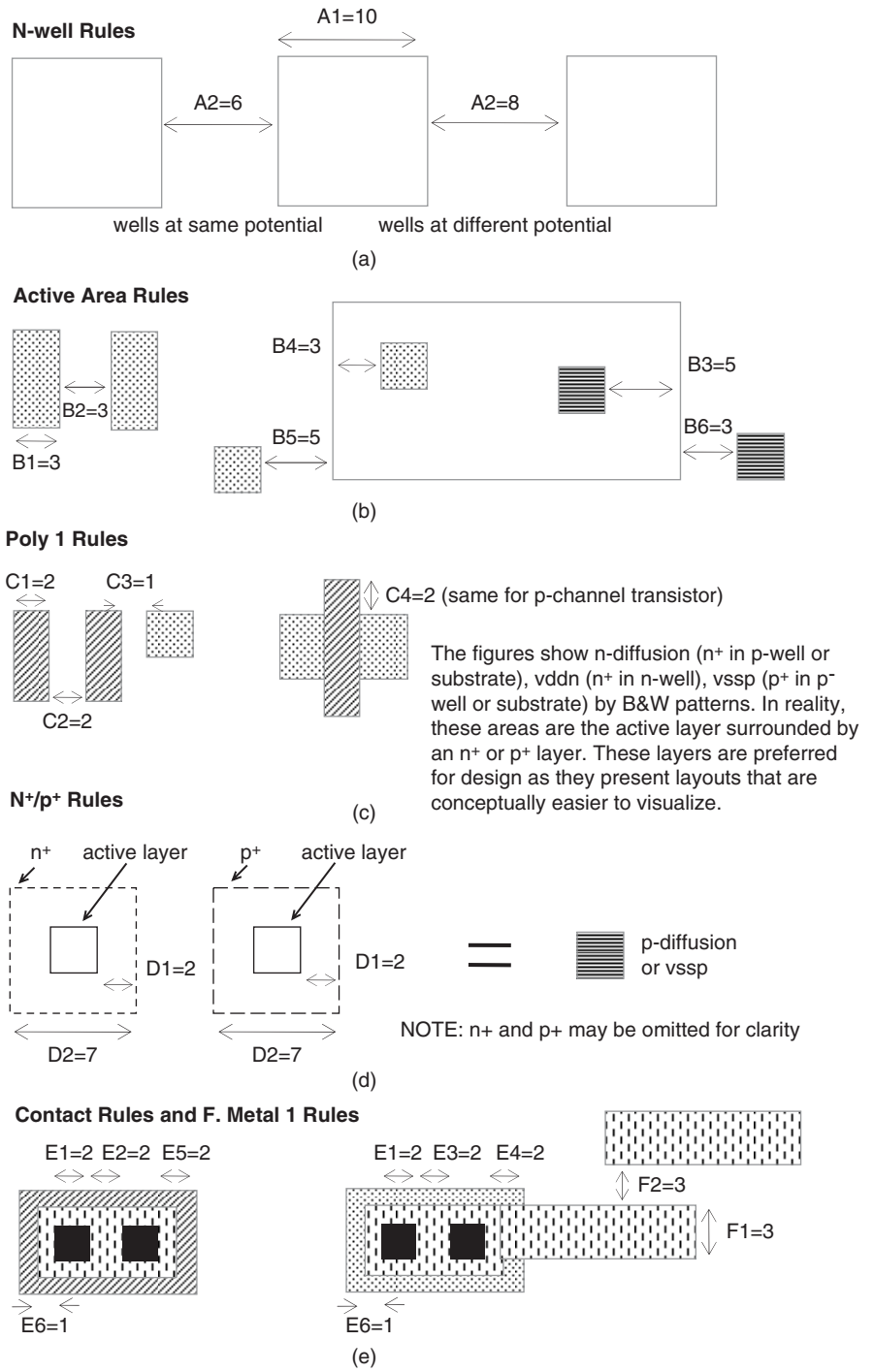


FIGURE 2.46

Continued

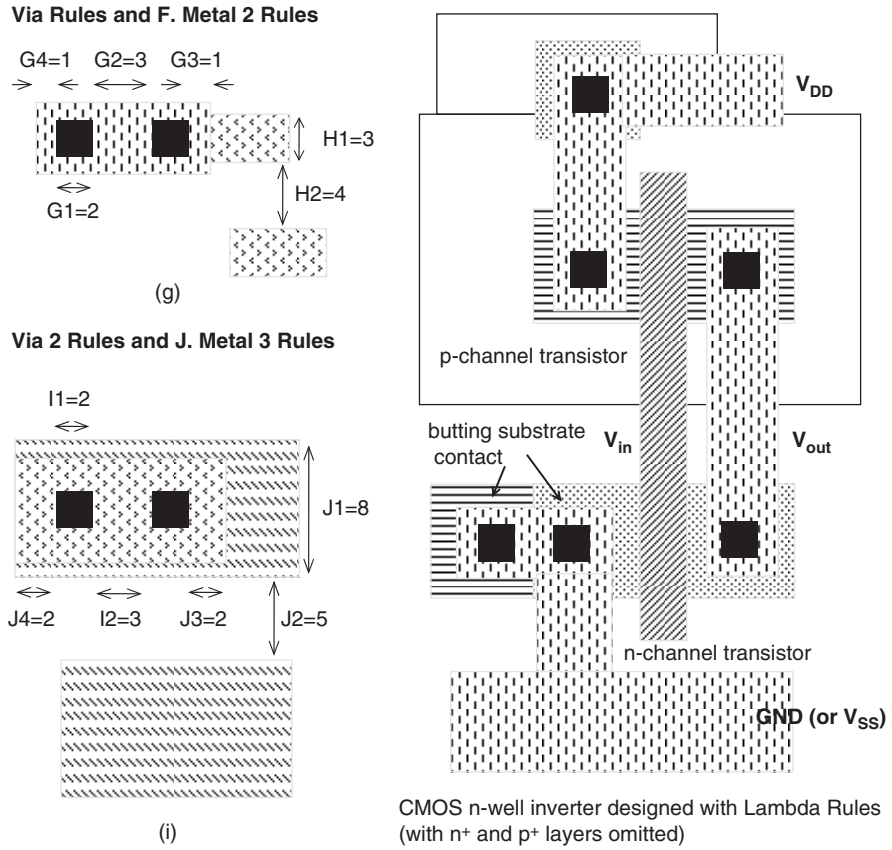


FIGURE 2.46

Illustration of layout rules with designated B&W patterns [Weste 1994].

tools have largely automated the floorplanning, placement, and routing tasks and processes, some designers still use stick diagrams in planning block layout designs and functional units.

2.5.3 **Layout design**

Although most of the chip-level physical layout design activities are done by running automated EDA tools, most physical layout design library cells (*a.k.a.* books) are still created and fine-tuned manually with the help of EDA tools such as a layout editor. In this subsection, we highlight a few physical layout design examples of small CMOS circuit blocks. The layer-overlapping color display seen on designers' computer screens is known as **symbolic layout**. A chip-level symbolic layout display is often called the **artwork**. Once a chip-level physical layout design is verified against engineering metrics (such as DRC, timing, yield)

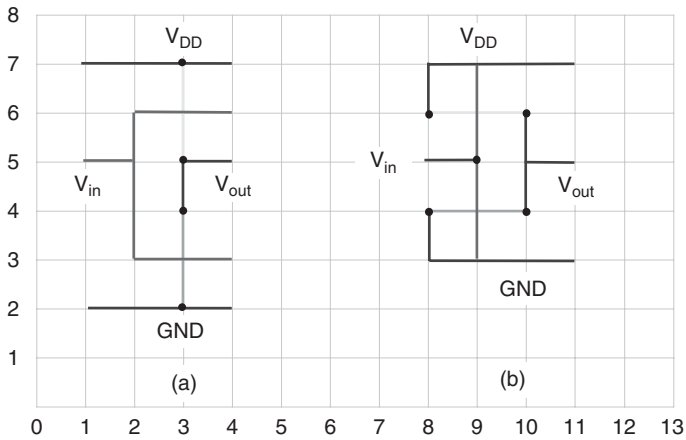


FIGURE 2.47

Stick diagrams for a CMOS inverter.

Table 2.6 Estimated Length on Stick Diagram X and Y Indexes

Index	Items	Length
For stick diagram of Figure 2.47a		
X(3)	$(4 + 1 + 2 + 1 + 4 + 2 + 4 + 1 + 2 + 1 + 4)\ \lambda$	26λ
Y(2)	$(2 + 4 + 2)\ \lambda$	8λ
Y(5)	$(2 + 2 + 2 + 4 + 2)\ \lambda$	12λ
Y(6)	$(2 + 2 + 2)\ \lambda$	6λ
Estimated block layout dimensions: 26 λ by 12 λ		
For stick diagram of Figure 2.47b		
X(9)	$(4 + 2 + 2 + 2 + 4 + 2 + 2 + 2 + 4)\ \lambda$	24λ
X(11)	$(4 + 2 + 4 + 2 + 4 + 2 + 4)\ \lambda$	22λ
Y(4)	$(4 + 1 + 2 + 1 + 4)\ \lambda$	12λ
Y(5)	$(2 + 4 + 2 + 2 + 2)\ \lambda$	12λ
Estimated block layout dimensions: 24λ by 12λ		

and approved, EDA tools are used to extract manufacturing mask data from the physical layout data for production masks.

Figure 2.48 shows a symbolic layout of a classic CMOS inverter that uses the n-well process. The layout design uses one metal layer. Typically, cells and blocks in a library have the same height so that wires for V_{DD} and GND can

be aligned precisely throughout a chip. With this CMOS inverter, space is left between the n-channel transistor and the p-channel transistor so that this inverter cell maintains the same height as the other cells to be described in this subsection. Note that, whenever possible, n-well contacts (with V_{DD}) are placed along the V_{DD} supply line, and substrate contacts are placed along GND. These contacts are necessary to provide good grounding for the well and the substrate. Once a cell is created manually, it is important to check for any physical layout design rule violations. Typically, EDA tools provide such a function known as a ***design rule check*** (DRC). It is important to note that, when performing DRC with an EDA tool, a correct rule set must be specified. For example, to check this CMOS inverter layout design for any DRC violations, the n-well–based design rule set must be specified in the application. Inappropriate use of design rule set would result in either not discovering or wrongly identifying DRC violations.

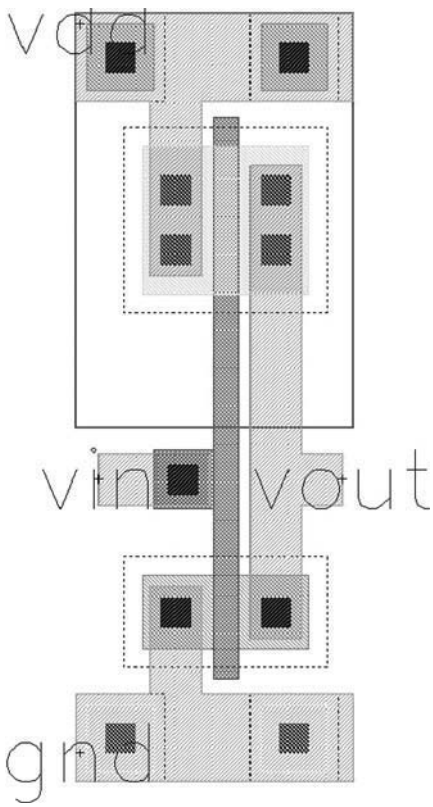


FIGURE 2.48

Symbolic layout of a CMOS inverter.

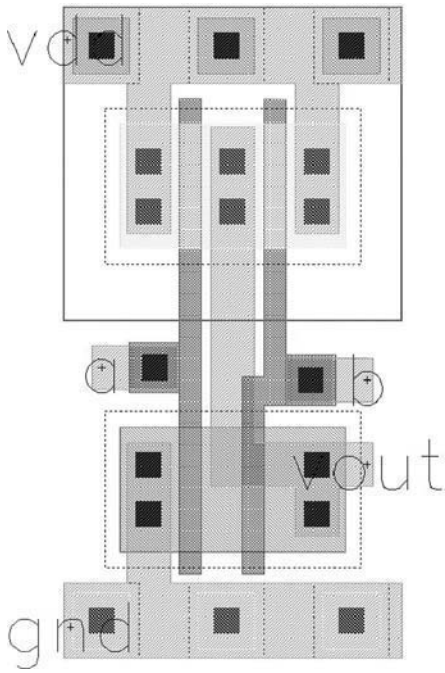


FIGURE 2.49

Symbolic layout of a 2-input 1-output CMOS NAND gate.

Figure 2.49 shows a symbolic layout for a 2-input NAND gate that uses one metal layer and the n-well process. Because of this limitation, its two inputs are accessed at different sides. Typically, library cells would have their inputs on one side and their outputs on the other side. This can effectively reduce the overall wire length when cells are used in functional blocks. When a second metal layer is available, input *b* in Figure 2.49 can easily be rerouted to the West along the side of input *a*.

Figure 2.50 shows a symbolic layout of a 3-input OR followed by a 2-input NAND block, which uses one metal layer and the n-well process. Because it also uses one metal layer, the inputs of the block are accessed from both sides, and the output goes out on the left side. When a second metal layer is available, one can reroute inputs to the West and the output to the East. As an alternative, the inputs can also be routed for access from the South by extending the Poly wires beyond GND.

Note that in Figure 2.50, the n-channel transistor controlled by input *a* is one third of the size of the p-channel transistors controlled by inputs *b*, *c,* and *d*. This is because the p-channel transistors of inputs *b*, *c,* and *d* are in series connection, and by the transistor equivalence theory, the equivalent transistor size

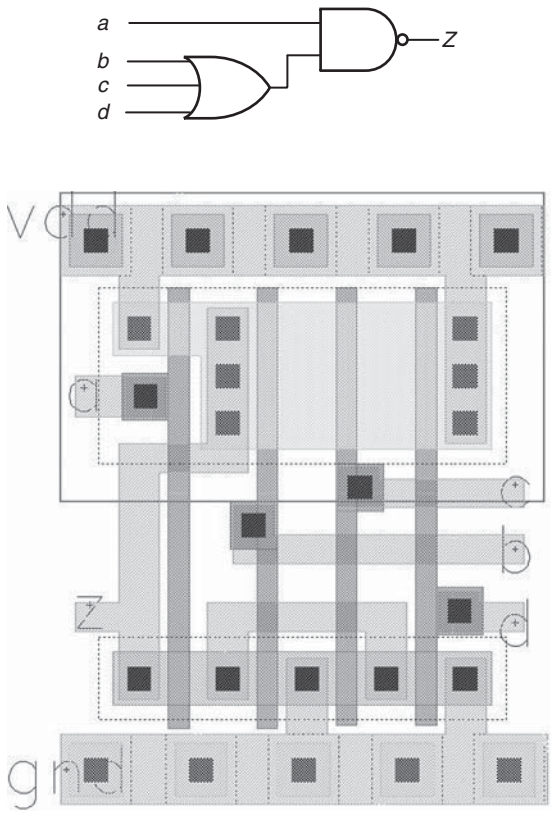


FIGURE 2.50

Symbolic layout of a 3-input-OR 2-input-NAND block.

of p-channel transistors controlled by inputs *b*, *c,* and *d* is the same as the size of p-channel transistor of input *a*.

Figure 2.51 shows a symbolic layout of grading-series transistors in an AND dynamic CMOS block [Weste 1994] with 4 inputs. The layout design uses transistors of varying sizes according to the position in the series structure to reduce delay. The n-channel transistor closest to the output is the smallest, with n-channel transistors increasing their size as they are placed nearer GND. The switching time is reduced, because there is less capacitance at the output. With older technologies, it provided 15% to 30% performance boost. However, with submicron technologies, this improvement is much less, at 2% to 4% in some cases. Nevertheless, the example demonstrates how layout designs of blocks can be optimized.

It is worth noting that often multiple techniques can be applied to a block. As an exercise, readers can attempt to improve the design of Figure 2.51 by first analyzing and identifying the problems associated with the design and then

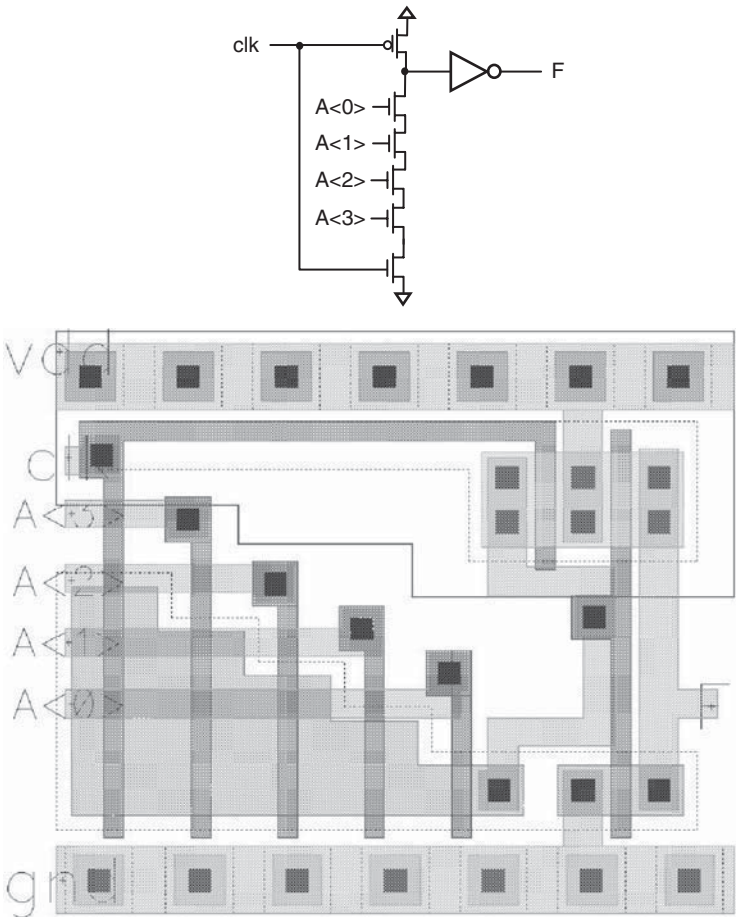


FIGURE 2.51

Symbolic layout of a 4-input AND gate by use of grading-series transistors. [Martin 2000].

modifying the circuit and layout designs that use the techniques discussed in this chapter to improve circuit speed, reduce transistor count, silicon area, and power consumption.

2.6 LOW-POWER CIRCUIT DESIGN TECHNIQUES

As mentioned earlier, there are three sources of power dissipation in CMOS circuits: dynamic power dissipation, short-circuit power dissipation, and static (leakage) power dissipation. Traditionally, dynamic power dissipation has been the dominant source of power dissipation. With continued scaling of CMOS

technology, however, leakage power dissipation has become a significant source of power consumption as well. This subsection describes some commonly used circuit-level techniques for reducing power dissipation.

2.6.1 **Clock-gating**

One commonly used technique to reduce power dissipation is to use **clock-gating**. The idea is that clock lines to circuits that are not being used are ANDed with a gate-control signal that disables the clock line to avoid unnecessary charging and discharging of unused circuits. Not all circuits are used at all times. Individual circuit use varies widely across applications and time, so there are many opportunities to use clock-gating.

The clock tree distributes the clock to sequential elements like flip-flops and latches, as well as to dynamic logic gates. Portions of the clock tree can be pruned by gating them with an AND gate as illustrated in Figure 2.52. When the gate-control signal is set to 0, it holds the clock line at a constant 0. This avoids charging and discharging of the capacitive load on the clock line and also prevents latches from changing state, thereby avoiding additional switching activity in any combinational logic being driven by the latch. For dynamic logic circuits, holding the clock at a constant 0 prevents the evaluate phase from occurring, thereby preventing the output from switching values. In practice, transparent latches are often used to gate clocks and prevent potential glitches that can happen with logic AND.

Clock-gating is effective at reducing dynamic power dissipation in unused sequential circuits and dynamic logic gates. Some limitations of clock-gating are that it does not prevent switching in static logic gates that may occur because of changes in the primary input values, and it does not reduce leakage power consumption. These limitations can be addressed by the use of power-gating.

2.6.2 **Power-gating**

Another way to reduce power dissipation in unused circuits is to use **power-gating** [Mutoh 1993; Sakata 1993]. The idea in power-gating is to switch off the power supply to unused circuits, thereby putting them in a "sleep" mode. This is typically implemented by having a gating transistor that can be turned off when the circuit is to be idle for an extended period of time. The gating

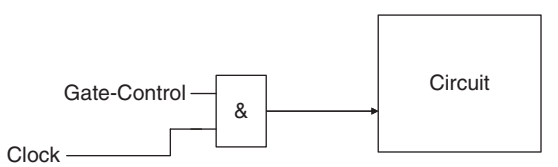


FIGURE 2.52

Clock-gating.

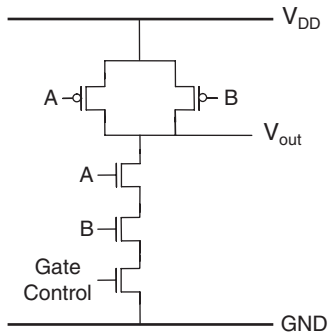


FIGURE 2.53

Power-gated 2-input NAND gate.

transistor can be either a header (p-channel transistor) or footer (n-channel transistor) transistor. Figure 2.53 illustrates a footer transistor. The gating transistor must be sized large enough to handle the amount of switching current at any given time so that there is no measurable amount of voltage drop across it. A footer transistor tends to require less area for a given switching current because of the higher mobility of electrons in an n-channel transistor compared with a p-channel header transistor. In a **multiple-V_T technology**, the gating transistor is typically implemented with a high V_T to minimize subthreshold leakage current through it. Power-gating can thus provide significant leakage power reduction, particularly when used in conjunction with circuits containing low V_T transistors.

Power-gating can be done at either a fine-grain or coarse-grain level. In **fine-grain power-gating**, the gating transistor is part of the standard cell logic. The advantage of this is that the burden of designing the gating transistor is left to the standard cell designer, and the cells can be easily handled by EDA tools. The drawback is that the gating transistor must be sized assuming worst-case conditions in which every cell is switching every clock cycle because nothing can be assumed about the module-level function. In **coarse-grain power-gating**, the gating transistor is part of the power distribution network rather than the standard cell and thus is shared among many gates. One advantage of this is that because only a fraction of the gates switch at any given time, the gating transistors can be sized smaller on aggregate compared with fine-grain power-gating. One issue for coarse-grain power-gating is that if too many gating transistors are switched simultaneously when going in and out of sleep mode, the current demand may overwhelm the power distribution network. Thus, some means for limiting the number of gating transistors that are simultaneously switched is needed.

Because the gating transistors are high V_{TH} devices, they can take several clock cycles to switch on and off and cause additional power dissipation. Thus, for power-gating to be efficient, the circuit must be idle for a sufficient number

of clock cycles so that the power savings justifies the time and cost of switching in and out of sleep mode.

When power-gating is implemented in sequential circuits, a means for retaining the sequential state is needed when the circuit goes into sleep mode. One simple approach is to scan the values in the storage elements into a memory before going into sleep mode, and then scan them back from the memory when the circuit wakes up.

Whereas clock-gating can only reduce dynamic power dissipation, power-gating can reduce both dynamic and leakage power dissipation. Because leakage power dissipation has become a sizable portion of overall power dissipation, power-gating has become a very important power reduction method. A drawback of power-gating compared with clock-gating is that it takes several clock cycles to switch in and out of sleep mode, and hence it is only efficient if the circuit will be idle for a sufficiently long time.

2.6.3 **Substrate biasing**

Another way to reduce leakage current (hence, leakage power dissipation) when a circuit is not being used is through **substrate biasing** [Seta 1995], which is also known as **variable threshold CMOS**. The idea is to adjust the threshold voltage by changing the substrate bias voltage (V_{SB}). Increasing the substrate bias voltage induces a body effect on the transistor that increases its threshold voltage (V_T). By having a substrate bias control circuit as illustrated in Figure 2.54, the substrate bias can be adjusted for normal operation to minimize V_T and maximize performance, and then when the circuit is in standby mode, the substrate bias can be adjusted to increase V_T to reduce the subthreshold leakage current. For example, the voltage on V_{Bp} could be set to V_{DD} in normal mode and $2V_{DD}$ in standby mode. The voltage on V_{Bn} could be set to 0 in normal mode and $-V_{DD}$ in standby mode. This would significantly reduce the leakage power dissipation.

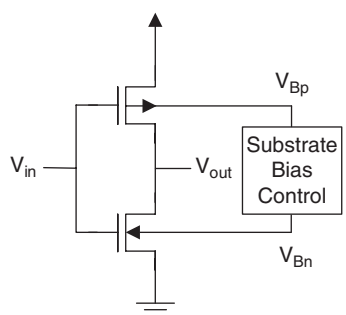


FIGURE 2.54

Substrate biasing.

One drawback of substrate biasing is that it requires a twin- or triple-well CMOS technology to apply different bias voltages to different parts of the chip. There is also a need to generate voltages outside of the normal 0 to V_{DD} power rail range that may require additional power pins on the chip.

2.6.4 Dynamic voltage and frequency scaling

The speed of a circuit depends linearly on the supply voltage. The idea in **dynamic voltage scaling** [Flautner 2001] is that during times when the circuit is not needing high performance, both its clock frequency and supply voltage can be scaled down. Because dynamic power dissipation depends on the square of the supply voltage and linearly on the frequency ($P = CV^2f$), if both the supply voltage and frequency are scaled down, there is a cubic reduction in power consumption.

Dynamic voltage scaling has been implemented in several commercial embedded microprocessors including the Transmeta Crusoe [Transmeta 2002], Intel Xscale [Intel 2003], and ARM IEM [ARM 2007]. When the processor is lightly loaded, the frequency and supply voltage are scaled down to save power, and when it is heavily executing, it is run at full frequency and voltage.

Figure 2.55 illustrates how a dynamic voltage–scaling scheme works. On the basis of the workload, the system requests a frequency change. First, the frequency is reduced, which takes on the order of hundreds of picoseconds, and then the voltage is ramped down, which takes on the order of hundreds of microseconds. Later, when switching back to high frequency, the voltage is first scaled back up to the normal voltage level, and then the frequency is raised back up.

Dynamic voltage scaling is a highly efficient way of reducing power consumption while still preserving functionality and meeting user expectations. It has been widely deployed.

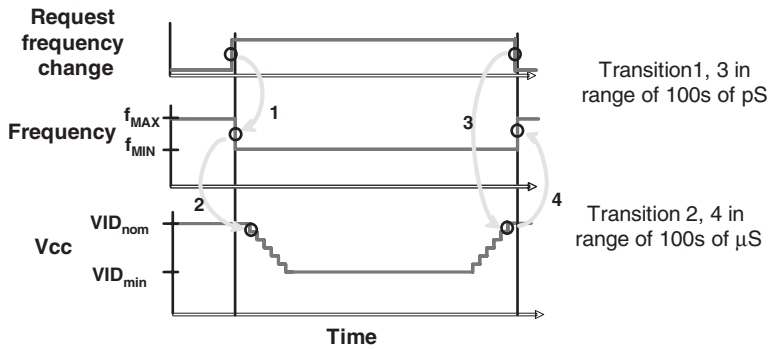


FIGURE 2.55

Dynamic voltage and frequency scaling.

2.6.5 **Low-power cache memory design**

Because microprocessor and ASIC chips contain cache memory often taking up more than half of the silicon space, power dissipation of these on-chip memory blocks can significantly contribute to the overall power consumption. In some cases, the static leakage power dissipation of cache memory contributes more than half of the chip's power consumption. Therefore, modern designs often use on-chip memory technologies with low-power features.

Power dissipation of on-chip memory blocks largely comes from the following functional units: the memory cells, the word and bit lines, and the peripheral circuits such as address decoders and sense amplifiers. In this subsection, we outline some of the low-power techniques applied with word and bit lines.

Figure 2.56 illustrates the memory cell of a typical on-chip cache SRAM memory block. A cell is being accessed (either READ or WRITE) by selected word and bit lines, which are connected to the outputs of address decoder circuits. The arrows indicate the leakage currents (because of bit lines being pre-charged to high) when the cell holds a 0 at the BL side and a 1 at the complementary side. For large on-chip memory, a word or bit line is a long interconnect that would connect to several thousands of cells. Longer word and bit lines not only require larger driving circuits at the outputs of address decoders but also cause concerns with respect to word/bit line delay and more power dissipation during word/bit line pre-charge.

To address these concerns, large on-chip memory is typically divided in many small sections so that each word or bit line drives a small number of cells. This technique is known as **banked cache design**. Both word and bit lines are also sectioned into a hierarchical structure such that each of the selected word

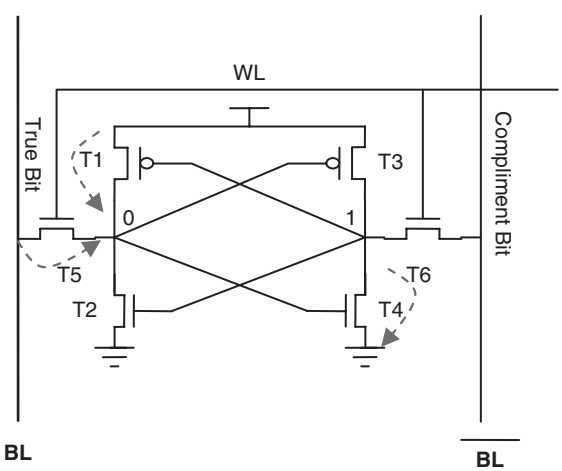


FIGURE 2.56

A typical SRAM cell.

and bit lines drives a few hundreds or fewer cells. A local sense amplifier bus is also used such that selected cache banks can connect to the nearest sense amplifiers, effectively reducing the length of active word and bit lines.

A technique known as **sub bit lines** [Karandikar 1998; Yong 2005] is illustrated in Figure 2.57. Each memory cell is connected to the main bit line by a sub bit line. A sub bit line is a short interconnect line that connects to a few cells. Only one selected sub bit line is connected to the main bit line at a time. Therefore, it significantly reduces the number of memory cells that load the main bit line at any time, which improves the bit-line response time. It also reduces leakage current, because inactive sub bit lines no longer need to be precharged. The disadvantage is that the addition of sub bit lines doubles the area used by bit line interconnects.

With multicore processor technologies becoming mainstream applications, more and more chips are making use of multi-port on-chip cache memory to maintain performance requirements. Classic hard-wired multi-port memory architecture usually uses dedicated word and bit lines to each memory cell for each port. Figure 2.58 illustrates a cell with 2 hard-wired ports. The addition of the second port not only increases the footprint of cache memory on silicon but also introduces additional leakage current (as indicated by arrows in Figure 2.58).

Figure 2.59 illustrates a new technique called **dynamic memory partitioning with isolation nodes** [Bajwa 2006, 2007; Chen 2007]. In theory, isolation nodes are placed on bit lines between neighboring memory cells. One port access is from the bottom of the bit line and the other port access is from the top of the bit line. When the two ports are accessing different cells, a selected isolation control line turns off the isolation nodes and divides the memory bank

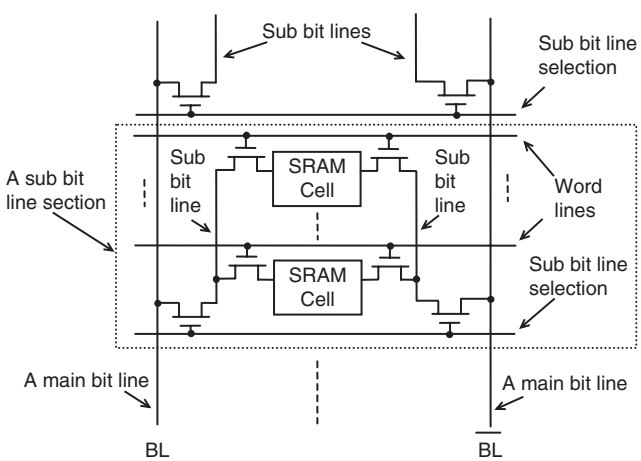


FIGURE 2.57

Illustration of sub bit lines.

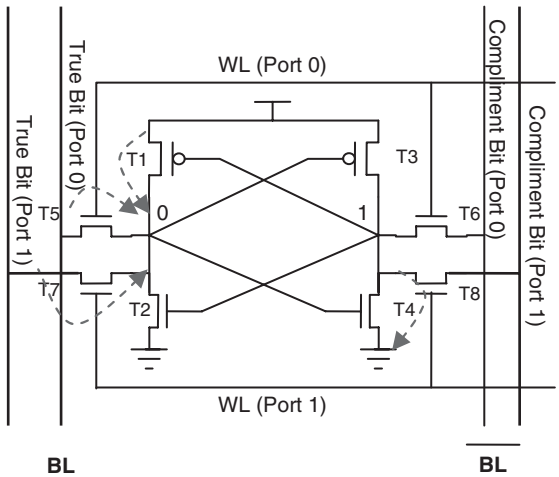


FIGURE 2.58

A typical hard-wired dual-port SRAM cell.

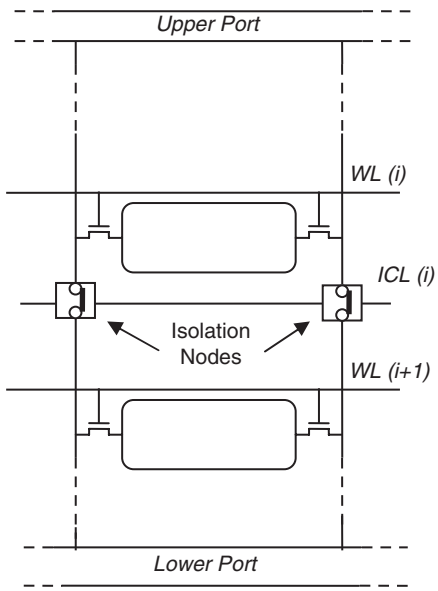


FIGURE 2.59

Illustration of energy-efficient and area-efficient dual-port SRAM.

into two virtually isolated sections to be accessed through the lower and upper ports. When the two ports are accessing the same memory location, all isolation nodes on the bit lines remain in the ON state.

One of the advantages of this dynamic memory partitioning technique that uses isolation nodes is the shared bit lines for the two ports. The length of active bit lines for both ports is shorter. Therefore, it reduces the silicon footprint of multi-port cache memory and improves bit-line response time. Another advantage is the low-power dissipation, because the shared bit line consumes no more power than the single-port configuration. In addition, leakage current remains the same as it is in a single-port configuration. This is because no dedicated bit lines and access transistors are used for the second port. By the use of local sense amplifiers and port multiplexing, this dynamic memory partitioning technique can be applied to on-chip cache memory with more than two ports. The same technique is applicable to DRAM. The disadvantage is that a port may need to pass through several isolation nodes to access a memory cell. The channel resistance of the pass transistors implementing the isolation nodes adds to the bit line response time. However, as the technology advances down to the 32-nanometer node and below, transistor channel resistance will become insignificant compared with wire resistance of the bit lines.

2.7 CONCLUDING REMARKS

CMOS technology has been the backbone of the many advances that have taken place in the past two decades, powering consumer appliances, automotives, personal and scientific computing, as well as many fascinating science and space explorations. Its advances have also made *electronic design automation* (EDA) tools possible and readily accessible to engineers. It is ironic that CMOS chips now power the computers on which engineers rely to design new chips. This chapter is intended to stimulate the reader's interest in the topic and provide background information for the reader to relate CMOS design to the EDA techniques to be discussed in the subsequent chapters.

New CMOS circuit technologies are still being developed. Currently, major improvements center on three fronts: transistors are used more efficiently to provide more computing and functionality, increasing circuit speed, and consuming less power. This chapter has provided some examples in all three of these improvements. For readers who wish to explore further on CMOS design, refer to more recent textbooks cited in the chapter and IEEE publications such as *IEEE Journal of Solid-State Circuits* (JSSC) and *IEEE International Solid-State Circuit Conference* (ISSCC).

2.8 EXERCISES

The following transistor parameters are used in **Exercises 2.1** to **2.13**:

For n-channel transistors:

$\mu_n C_{ox} = 190\ \mu A/V^2$
$C_{ox} = 3.4 \times 10^{-3}\ pF/(\mu m)^2$
$V_{tn} = 0.7\ V$
$r_{ds}(\Omega) = 5000\ L(\mu m)/I_D(mA)$ $---$ in active region
$C_j = 5 \times 10^{-4}\ pF/(\mu m)^2$
$C_{j-sw} = 2.0 \times 10^{-4}\ pF/\mu m$
$C_{gs(overlap)} = C_{gd(overlap)} = 2.0 \times 10^{-4}\ pF/\mu m$

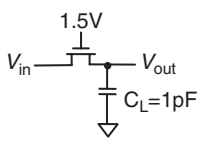
For p-channel transistors:

$\mu_p C_{ox} = 50\ \mu A/V^2$
$C_{ox} = 3.4 \times 10^{-3}\ pF/(\mu m)^2$
$V_{tn} = -0.8\ V$
$r_{ds}(\Omega) = 6000\ L(\mu m)/I_D(mA)$ $---$ in active region
$C_j = 6 \times 10^{-4}\ pF/(\mu m)^2$
$C_{j-sw} = 2.5 \times 10^{-4}\ pF/\mu m$
$C_{gs(overlap)} = C_{gd(overlap)} = 2.0 \times 10^{-4}\ pF/\mu m$

2.1. (Integrated-Circuit Technology) An n-channel (or p-channel) transistor in the active region is measured to have $I_D = 20\ \mu A$ when $V_{DS} = V_{eff}$. As V_{DS} increases by 0.5 V, I_D increases to 23 μA, estimate the output impedance r_{ds}.

2.2. (Integrated-Circuit Technology) Estimate the capacitances C_{gs}, C_{gd}, C_{db}, and C_{sb} for an n-channel transistor and a p-channel transistor with $W = 10\ \mu m$ and $L = 1.2\ \mu m$, assuming the junction areas A_s (at the source) and A_d (at the drain) are 40 $(\mu m)^2$ and the perimeter of each (P_s and P_d) is 12μm.

2.3. (Integrated-Circuit Technology) Consider the circuit below, when V_{in} is 1.2 V. Estimate V_{out} when the n-channel pass transistor ($W = 2.4\ \mu m$ and $L = 1.2\ \mu m$) is turned ON.



2.4. (Integrated-Circuit Technology) The effects of technology scaling are outlined in the following table. Now assume that all dimensions are scaled by S, but the voltage and doping levels are only scaled by $\sqrt{S}$, and estimate the scaling factor for other parameters listed in the Table 2.7.

2.5. (CMOS Logic) Design a CMOS circuit that implements $F = a \cdot b \cdot \bar{c} + \bar{a} \cdot c \cdot d$. Choose transistor sizes to give equal rise and fall times at the output.

Table 2.7 Effects of Scaling

Parameter	Scaling Factor
Device dimensions (t_{ox}, L, W, junction depth)	$1/S$
Doping concentration	S
Voltage	$1/S$
Current	$1/S$
Capacitance	$1/S$
Delay time	$1/S$
Power dissipation (per gate)	$1/S^2$
Power-delay product	$1/S^3$

2.6. (CMOS Logic) Design a circuit that converts 5.0 V TTL logic outputs to a CMOS logic block that uses a 3.3 V power supply.

2.7. (CMOS Logic) Design a circuit that interfaces the outputs of a 1.3 V CMOS logic block with the inputs of a 3.3 V CMOS block.

2.8. (CMOS Logic) Consider the circuit design in **Exercise 2.5** and analyze and estimate the static power dissipation. Also, assuming the circuit block switches at 5 MHz, estimate the dynamic power dissipation.

2.9. (Advanced Integrated-Circuit Design) Design a 2-input differential AND/NAND circuit block. Specify individual transistor sizes such that the rise and fall times at each output are roughly the same. Assume $V_{DD} = 3.3$ V and an external $C_L = 1$ pF is at each output.

2.10. (CMOS Physical Design) Construct a stick diagram of a transmission-gate and inverter-based D latch. Draw the transistor schematic first.

2.11. (CMOS Physical Design) Construct a stick diagram of a single-bit full-adder by first drawing its transistor schematic.

2.12. (CMOS Physical Design) Use a layout editor to design a physical layout for the D latch shown in Figure 2.21.

2.13. (CMOS Physical Design) Use a layout editor to design a physical layout for the single-bit carry circuit shown in Figure 2.23.

2.14. (CMOS Physical Design) Analyze the circuit block and layout design in Figure 2.51. Identify further improvements. Improve the circuit block by use of the techniques discussed in this chapter. Use an EDA layout editor to modify the original layout design by use of the same n-well process.

2.15. (Low-Power Design) List the advantages and disadvantages of power-gating versus clock-gating.

2.16. (Low-Power Design) Describe the advantages and disadvantages of substrate biasing.

ACKNOWLEDGMENTS

We thank Wan-Ping Lee, Guang-Wan Liao, and Professor Yao-Wen Chang of National Taiwan University for helping with generating the symbolic layouts, and Andrew Wu, Meng-Kai Hsu, and Professor James C.-M. Li for reviewing the manuscript. We also thank Professor Eric MacDonald of University of Texas at El Paso and Professor Martin Margala of University of Massachusetts at Lowell for their constructive comments and suggestions.

REFERENCES

R2.0 Books

[Karim 2007] M. Karim and X. Chen, *Digital Design: Basic Concepts and Principles,* CRC Press, New York, 2007.

[Martin 2000] K. Martin, *Digital Integrated Circuit Design,* Oxford University Press, New York, 2000.

[Rabaey 2003] J. M. Rabaey, A. Chandrakasan, and B. Nikolić, *Digital Integrated Circuits: A Design Perspective,* Second Edition, Prentice-Hall, Englewood Cliffs, NJ, 2003.

[Wakerly 2001] J. F. Wakerly, *Digital Design: Principles and Practices,* Third Edition, Prentice-Hall, Englewood Cliffs, NJ, 2001.

[Weste 1994] N. H. E. Weste and K. Eshraghian, *Principles of CMOS Design—A System Perspective,* Second Edition, Addison-Wesley, Reading, MA, 1994.

R2.6 Low-Power Design

[ARM 2007] ARM Ltd., 1176JZ(F)-S Documentation, http://www.arm.com/products/CPUs/ARM1176.html, 2007.

[Bajwa 2006] H. Bajwa and X. Chen, Area-efficient dual-port memory architecture for multi-core processors, in *Proc. Junior Scientists Conf.*, pp. 49–50, April 2006.

[Bajwa 2007] H. Bajwa and X. Chen, Low-power high-performance and dynamically reconfigured multi-port cache memory architecture, in *Proc. IEEE Int. Conf. on Electrical Engineering*, April, 2007.

[Chen 2007] X. Chen and H. Bajwa, Energy-efficient dual-port cache architecture with improved performances, Institution of Engineering and Technology. *J. of Electronics Letters*, 43(1), pp. 12–13, January, 2007.

[Flautner 2001] K. Flautner, S. Reinhardt, and T. Mudge, Automatic performance setting for dynamic voltage scaling, in *Proc. Int. Conf. on Mobile Computing and Networking*, pp. 260–271, May 2001.

[Intel 2003] Intel Corp., Intel Xscale Core Developer's Manual, http://developer.intel.com/design/intelxscale/, 2003.

[Karandikar 1998] A. Karandikar and K. K. Parhi, Low power SRAM design using hierarchical divided bitline approach, in *Proc. Int. Conf. Computer Design*, pp. 82–88, October 1998.

[Mutoh 1993] S. Mutoh, T. Douseki, Y. Matsuya, T. Aoki, and J. Yamada, 1V high-speed digital circuits technology with 0.5 μm multi-threshold CMOS, in *Proc. IEEE Int. ASIC Conf.*, pp. 186–189, September 1993.

[Sakata 1993] T. Sakata, M. Horiguchi, and K. Itoh, Subthreshold-current reduction circuits for multi-gigabit DRAM's, in *Proc. Symp. on VLSI Circuits*, pp. 45–46, May 1993.

[Seta 1995] K. Seta, H. Hara, T. Kuroda, M. Kakumu, and T. Sakurai, 50% active-power saving without speed degradation using standby power reduction (SPR) circuit, *Proc. Int. Solid-State Circuits Conf.*, pp. 318–319, February 1995.

[Transmeta 2002] Transmeta Corp., *Crusoe Processor Documentation*, http://www.transmeta.com, 2002.

[Yong 2005] B. D. Yong and L.-S. Kim, A low power SRAM using hierarchical bit line and local sense amplifier, *IEEE J. Solid-State Circuits*, 40(6), pp. 1366–1376, June 2005.